A Hierarchical Fuzzy System with High Input Dimensions for Forecasting Foreign Exchange Rates

France Cheong

Abstract—Fuzzy systems suffer from the curse of dimensionality as the number of rules increases exponentially with the number of input dimensions. Although several methods have been proposed for eliminating the combinatorial rule explosion, none of them is fully satisfactory as there are no known fuzzy systems that can handle a large number of inputs so far. In this paper, we describe a method for building fuzzy systems with high input dimensions based on the hierarchical architecture and the MacVicar-Whelan meta-rules. The proposed method is fully automated since a complete fuzzy system is generated from sample input-output data using an Evolutionary Algorithm. We tested the method by building fuzzy systems for two different applications, namely the forecasting of the Mexican and Argentinian pesos exchange rates. In both cases, our approach was successful as both fuzzy systems performed very well.

I. INTRODUCTION

Universal function approximators are a class of algorithms capable of approximating any real continuous function on a compact set to arbitrary accuracy. Given a function \( F : R^n \rightarrow R \) which is continuous, there is a universal function approximator which can approximate \( F \) uniformly on compact subset \( R^n \) to any degree of accuracy. Model-free estimators can estimate a function without a mathematical description of how the input depends on the output. Instead of a mathematical model, data samples are used to train the estimator to estimate the function in question. Model-free estimators are becoming very popular because with the increasing complexity of systems, our ability to mathematically model them is greatly reduced. Both Fuzzy Logic (FL) and Artificial Neural Networks (ANNs) exhibit universal approximation and model-free properties.

Fuzzy logic is based on Zadeh’s [1–4] theory of fuzzy sets. It is a superset of conventional boolean or crisp logic that has been extended to handle the concept of partial truth i.e. truth values between completely true and completely false. It allows the use of soft linguistic variables on a continuous range of truth values (known as membership values). This allows intermediate values to be defined between conventional binary values. Fuzzy logic resembles human reasoning in its use of approximate information and uncertainty to generate decisions. It was specifically designed to mathematically represent uncertainty and vagueness and provide formalized tools for dealing with the imprecision intrinsic to many problems. Fuzzy systems can be built either crudely using expert knowledge or more accurately using historical input-output data in conjunction with a search algorithm such as an Evolutionary Algorithm (EA). It must be noted that only an optimized fuzzy system can exhibit universal function approximation capabilities as there is no way a system built using expert knowledge can exhibit this property.

Using high input dimensions in ANNs is not a problem, however, this is not the case with fuzzy systems. In conventional fuzzy systems, the number of rules is of order \( O(T^k) \), where \( T \) is the number of fuzzy sets per dimension and \( k \) the number of dimensions. Due to the exponential complexity of fuzzy systems, the number of rules (as well as the computational complexity) increases exponentially with the number of dimensions. For example, if the ANN application reported in [5] is implemented using a conventional fuzzy system, that is, with 64 dimensions, 3 fuzzy sets per dimension (5 or 7 fuzzy sets per dimension is more common), there would be \( 3^{64} \) rules in the rule base i.e. 3,433 trillion rules.

Although several methods have been proposed in the fuzzy logic research literature for eliminating the combinatorial rule explosion, none of them is fully satisfactory since there is no known fuzzy system that can handle 64 inputs so far. Given the current situation, our aim in this study is to develop a methodology for designing fuzzy systems with large number of inputs. The proposed methodology will be fully automated since we propose to automatically generate a complete fuzzy system from sample input-output data.

II. RELATED WORK

Several approaches have been proposed to address the combinatorial rule explosion problem:

1) Hierarchical fuzzy system. Hierarchical fuzzy systems (HFS) can be used to reduce the number of rules from an exponential function of system variables to a linear one. An HFS is built by dividing a global task into sub-tasks, designing independent fuzzy systems for each sub-task, and devising a strategy for coordinating the fuzzy sub-systems to achieve the global objective. Raju et al. [6, 7] proposed a hierarchical structure (shown in Figure 1) in which the most influential system variables are at the first level, the next most important variables at the next level, and so on. The first-level sub-system generates an approximate output which is modified by the second-level sub-system; the inputs to the second-level sub-system being the output from the first-level sub-system and a system variable. This process is repeated at succeeding levels of the hierarchy to eventually produce an output. No automation of the design of the fuzzy system was involved. The main problem with hierarchical fuzzy systems is the loss
of linguistic interpretability of the overall model [8]. This is due to the fact that the (artificial) intermediate variables of the model are usually not directly related to the variables of the actual system and consequently, they do not have any physical or linguistic meaning. As a result, the fuzzy rules in the middle of the hierarchical structure are difficult to design and this phenomenon becomes prominent as the number of levels in the HFS grows. Since Raju’s work on HFS, there have been other work in that area, however, they all tend to suffer from the same drawback.

2) Rule reduction using GAs [9], ANNs [10] and clustering [11]. Although rule reduction is a technique that works, there is no guarantee that the reduced rule base will generate an output for every possible combination of the inputs as some rules might be missing.

3) Using a disjunctive form of the conjunctive rules. In [12], an attempt to eliminate the curse of dimensionality was made by providing a disjunctive form of the conjunctive rules commonly used in a fuzzy system. However, a comment [13] on this paper proved the approach taken to be mathematically invalid.

4) Using a function rather than rules! The approach taken in [14] was simply to replace the rule base with an “adaptive analytic” function that computes the output of the fuzzy system. However, depriving the fuzzy system of its rule base causes the system to loose the interpretability of its reasoning mechanism. Furthermore, such a system was not proved to retain the universal approximation property.

5) New fuzzy system. A new fuzzy system was proposed in [15] whose number of parameters grows linearly depending upon the number of inputs. The authors also used the Stone-Weierstrass theorem to prove that the new fuzzy system possesses the universal approximation property. Although the fuzzy system can be used with high number of inputs, it was only tested on a system with 2 inputs.

III. METHODOLOGY

A. Constrained optimization technique

When automating the design of fuzzy systems using Evolutionary Algorithms, very often the resulting design is not interpretable because of the formation of messy fuzzy sets and scrambled rule bases [16]. Recently, the interpretability of fuzzy systems has been the subject of numerous researches. In [16–18], we constrained the optimization of fuzzy logic controllers (FLCs) used in engineering applications (fuzzy systems are known as fuzzy logic controllers (FLCs) in control systems) by using the MacVicar-Whelan rule base [19]. This is a standard template rule base built according to common engineering sense and experience with fuzzy logic. It defines a reasonable set of rules that can be adjusted by excluding, modifying or adding new control rules based on the specificity of the control problem. If the input variables of the FLC are the error and the change in error and the output variable the change in control output, then the template rule base can be formulated according to the following MacVicar-Whelan meta-rules:

1) If both the error and the change in error are zero, then the change in output is zero.
2) If the error is tending towards zero at a satisfactory rate, then the change in output is zero.
3) If the error is not self-correcting, then the change in output is not zero and depends on the sign and magnitude of the error and change in error.

Three examples of rule bases expressed as a fuzzy associative matrices (FAM) [20] formulated using the MacVicar-Whelan’s meta-rules bases are shown in Figure 2: a 3 by 3, a 5 by 5 and a 7 by 7 rule base. In Figure 2(a), the rows of the matrix represent the three fuzzy sets of variable $x_1$ labelled as 1, 2 and 3 while the columns represent the three fuzzy sets of variable $x_2$.

The cells of the matrix represent the fuzzy sets of the output of the FLC which are labelled as 1,2 and 3. The rules in the 3 by 3 rule base are interpreted in rowwise order as shown in Figure 3.

B. Hierarchical architecture

Our method for automating the construction of fuzzy systems with high input dimensions is based on the hierarchical configuration architecture. We use simple fuzzy logic controllers (FLCs) with two inputs and a single output as building blocks for hierarchical fuzzy logic controllers (HFLCs) similar to Raju’s architecture shown in Figure 1. In order to make the rules used by the sub-controllers interpretable, we use the MacVicar-Whelan meta-rules. Although, these meta-rules have successfully been applied to control systems, we believe that they can adapted for use in other areas as well. There are several ways to do this. One way of doing this would be to extend the meta-rules to cope with new situations. However, we believe that a simpler and more effective approach is to design the sub-controllers using the original MacVicar-Whelan meta rules and customize them for the particular situation by modifying the sign of the
inputs (to take care of the direction of correlation of the input with the output) and search for the most appropriate membership functions using an Evolutionary Algorithm (EA) and sample input-output data. This is a novel approach for constructing fuzzy systems with high input dimension that are interpretable as well as accurate since they are optimized using sample input-output data.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Condition</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$x_1 = 1$ AND $x_2 = 1$</td>
<td>$output = 1$</td>
</tr>
<tr>
<td>2</td>
<td>$x_1 = 1$ AND $x_2 = 2$</td>
<td>$output = 1$</td>
</tr>
<tr>
<td>3</td>
<td>$x_1 = 1$ AND $x_2 = 3$</td>
<td>$output = 2$</td>
</tr>
<tr>
<td>4</td>
<td>$x_1 = 2$ AND $x_2 = 1$</td>
<td>$output = 1$</td>
</tr>
<tr>
<td>5</td>
<td>$x_1 = 2$ AND $x_2 = 2$</td>
<td>$output = 2$</td>
</tr>
<tr>
<td>6</td>
<td>$x_1 = 2$ AND $x_2 = 3$</td>
<td>$output = 3$</td>
</tr>
<tr>
<td>7</td>
<td>$x_1 = 3$ AND $x_2 = 1$</td>
<td>$output = 2$</td>
</tr>
<tr>
<td>8</td>
<td>$x_1 = 3$ AND $x_2 = 2$</td>
<td>$output = 3$</td>
</tr>
<tr>
<td>9</td>
<td>$x_1 = 3$ AND $x_2 = 3$</td>
<td>$output = 3$</td>
</tr>
</tbody>
</table>

Mutation is performed either on the best individual found so far in the evolution process or on any randomly drawn individual from the population. The selected individual is mutated by adding a perturbation vector to it which is calculated as the scaled difference between two randomly sampled individuals from the population. Mathematically this can be expressed as: $X_a = X_b + \alpha(X_c - X_d)$, where $\alpha$ is a system parameter.

Offsprings (trial individuals) are created using a recombination process whereby their genes are selected either from the target individual or the mutant individual. Two methods of recombination are used in DE: binary and exponential. In binomial recombination, a series of binomial experiments are conducted to determine the parent from which the gene will come from. Each experiment is mediated by a crossover constant, $CR$, where $0 \leq CR \leq 1$. Starting from a randomly selected gene, the source of each gene is determined by comparing $CR$ to a uniformly distributed random number from the interval $[0, 1)$. If the random number is greater than $CR$, the offspring obtains its gene from the target individual, otherwise, the gene comes from the mutant individual.

In exponential recombination, a single contiguous block of genes of random size and location is copied from the mutant individual to a copy of the target individual to produce an offspring. Starting from a randomly selected gene, genes are copied from the mutant individual while a uniformly distributed random number from the interval $[0, 1)$ is less than $CR$.

**C. Differential Evolution**

Differential Evolution (DE) [21] was selected as the Evolutionary Algorithm as we found it to work very well for real-valued optimizations in previous studies. DE is different to other EAs in that it uses population-derived noise to adapt the mutation rate of the evolution process and it is simple and very fast. In DE, individuals are represented as real-valued vectors.

For each generation of the evolution process, each individual (known as target individual) of the population competes against a new individual (known as trial individual) for survival to the next generation and only the fitter of the two individuals survives. The trial individual is created by recombining the target individual with another individual created by mutation (known as mutant individual).
the neighbouring fuzzy sets. Furthermore, since we use a universe of discourse normalized to the range [0.0, 1.0] and we avoid the use of trapezoidal fuzzy sets for the first and last fuzzy set, we fix the apices of the first and last fuzzy sets to 0.0 and 1.0 respectively. Our scheme for representing the membership functions results in a greatly reduced number of parameters. If we use five fuzzy sets per variable as shown in Figure 4, we only need three parameters to represent the membership functions and if we use seven fuzzy sets then we need five parameters.

IV. EXPERIMENTATION AND RESULTS

We experimented with the proposed method by constructing two HFLCs for time series forecasting.

A. Experiment 1: Learning the Mexican pesos foreign exchange rate

In the first experiment, we built an HFLC that learnt the relationship between 9 inputs and the foreign exchange rate of the Mexican pesos per US dollar. The 9 inputs used were: representative interest rate, funding interest rate, commercial banks average cost of funds, gross domestic product, export price index, import price index, terms of trade, monetary uses and international reserves. The data was obtained from a commercial database and covered the period Jan-00 to Dec-03 at a monthly frequency.

With 9 inputs configured using an architecture similar to that shown in Figure 1, the HFLC for forecasting the Mexican pesos exchange consists of 8 FLCS. The inputs to the HFLC were configured as follows: \(x_0\) = representative interest rate, \(x_1\) = funding interest rate, \(x_2\) = commercial banks average cost of funds, \(x_3\) = gross domestic product, \(x_4\) = export price index, \(x_5\) = import price index, \(x_6\) = terms of trade, \(x_7\) = monetary uses, and \(x_8\) = international reserves.

Based on knowledge derived from finance, we determined all inputs to vary directly with the output (an increase in the input causes an increase in the rate of exchange) except for \(x_3\) and \(x_4\) which were inversely related (an increase in the input causes a decrease in the rate of exchange). This information regarding the direction of correlation was used to manually configure the sign of the inputs since we used the standard MacVicar-Whelan meta-rules to design a single type of rule base for all inputs (we tried to use the evolutionary algorithm to automatically determine the direction of correlation and we were not successful as it was having a hard time to do so).

If we use 5 fuzzy sets per input/output variable, we need a vector of 72 (8 FLCS x 3 variables per FLC x 3 membership function parameters per variable) parameters to represent a candidate HFLC solution. DE is used to evolve a population of candidate solutions (or individuals) until an acceptable solution is found.

The strategy used for mutation and recombination was based on the best individual found so far in the evolution process rather than a random individual. Mutation was performed by adding a perturbation vector to the best individual while recombination consisted of creating a trial individual by conducting a series of binomial experiments to determine whether to inherit genes from the target or mutant individual.

We used a population of 100 individuals and DE system parameters \(\alpha = 0.7\), and \(CR = 0.7\). The objective function used to guide the evolution process was the sum of squares of the errors (SSE) between the actual time series and the forecasted one. The learning process was performed within 200 generations and as can be seen from Figure 5, the time series was closely approximated. We experimented with 5 by 5, 7 by 7 and 9 by 9 rule bases for all sub-controllers and found a 5 by 5 rule base to perform very well.

We compared the performance of the HFLC with a random walk which assumes that the best estimate of the next value is the current value of the variable. The random walk is often used as benchmark for comparing the performance of forecasting systems as it is very hard to beat. The sum of squares of the errors (SSE) of the HFLC was 2.1 as compared to 1.7 for the random walk method. Figure 5 shows the actual time series, the HFLC-approximated time series and the time series generated by the random walk method.

B. Experiment 2: Learning the Argentinan pesos foreign exchange rate

In a second experiment, we built an HFLC similar to the first one, but this time for learning the relationship between the same 9 inputs and the foreign exchange rate of the Argentinan pesos per US dollar. The data obtained covered the period Jan-96 to Dec-03 at a monthly frequency. As for the Mexican case, the learning process was performed within 200 generations using DE and the time series was closely reproduced. The SSE of the HFLC was better than that of the random walk as it was 1.1 as compared to 3.5. The results of the second experiment are shown in Figure 6.

V. CONCLUSION

We have presented an approach for building fuzzy systems with high input dimensions based on the hierarchical architecture paradigm and the MacVicar-Whelan meta-rules. We have tested the approach by building two fuzzy systems: one for forecasting the exchange rate of the Mexican pesos and another one for forecasting the exchange rate of the Argentinan pesos. In both cases, the approach was successful as the outputs of the fuzzy systems closely approximate the financial time series. Both fuzzy systems used 9 inputs and 5
fuzzy sets per inputs. Fuzzy systems built using our approach are both optimized and interpretable as they are built using sample input-output data and the MacVicar-Whelan meta rules. In the future, we shall further confirm the feasibility
of our approach by experimenting with even higher input dimensions as the approach is scalable to any number of inputs.

REFERENCES