Learning Fuzzy Rules with Evolutionary Algorithms — an Analytic Approach

Jens Kroeske[1], Adam Ghandar[2], Zbigniew Michalewicz[3], and Frank Neumann[4]

[1] School of Mathematics, University of Adelaide, Adelaide, SA 5005, Australia
[2] School of Computer Science, University of Adelaide, Adelaide, SA 5005, Australia
[3] School of Computer Science, University of Adelaide, Adelaide, SA 5005, Australia; also at Institute of Computer Science, Polish Academy of Sciences, ul. Ordona 21, 01-237 Warsaw, Poland, and Polish-Japanese Institute of Information Technology, ul. Koszykowa 86, 02-008 Warsaw, Poland
[4] Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany

Abstract. This paper provides an analytical approach to fuzzy rule base optimization. While most research in the area has been done experimentally, our theoretical considerations give new insights to the task. Using the symmetry that is inherent in our formulation, we show that the problem of finding an optimal rule base can be reduced to solving a set of quadratic equations that generically have a one dimensional solution space. This alternate problem specification can enable new approaches for rule base optimization.

1 Introduction

Fuzzy rule based solution representations combined with evolutionary algorithms are a powerful real world problem solving technique, for example see [2, 5, 9, 15, 18, 19]. Fuzzy logic provides benefits in naturally representing real world quantities and relationships, fast controller adaptation, and a high capacity for solutions to be interpreted. The typical scenario involves using an evolutionary algorithm to find optimum rule bases with respect to some application specific evaluation function, see [7, 11, 13].

A fuzzy rule is a causal statement that has an *if-then* format. The *if* part is a series of conjunctions describing properties of some linguistic variables using fuzzy sets that, if observed, give rise to the *then* part. The *then* part is a value that reflects the consequence given the case that the *if* part occurs in full. A rule base consists of several such rules and is able to be evaluated using fuzzy operators to obtain a value given the (possibly partial) fulfilment of each rule.

Membership functions are a crucial part of the definition as they define the mappings to assign meaning to input data. They map crisp input observations of linguistic variables to degrees of membership in some fuzzy sets to describe properties of the linguistic variables. Suitable membership functions are designed depending on the specific characteristics of the linguistic variables as well as peculiar properties related to their use in optimization systems. Triangular membership functions are widely used primarily for the reasons described in [16].

Other common mappings include 'gaussian' [11] and 'trapezoidal' [8] membership functions. The functions are either predefined or determined in part or completely during an optimization process. A number of different techniques have been used for this task including statistical methods [7], heuristic approaches [2], and genetic and evolutionary algorithms [5, 9, 14, 18]. Adjusting membership functions during optimization is discussed in [9, 20].

A financial computational intelligence system for portfolio management is described in [7]. Fuzzy rule bases are optimized in an evolutionary process to find rules for selecting stocks to trade. A rule base that could be produced using this system could look as follows:

– If *Price to Earnings Ratio* is *Extremely Low* then *rating* = 0.9
– If *Price Change* is *High* and *Double Moving Average Sell* is *Very High* then *rating* = 0.4

The *if* part in this case specifies some financial accounting measures (Price to Earnings ratio) and technical indicators [1] used by financial analysts; the output of the rule base is a combined rating that allows stocks to be compared relative to each other. In that system rule bases were evaluated in the evolutionary process using a function based on a trading simulation.

The task of constructing rule base solutions includes determining rule statements, membership functions (including the number of distinct membership sets and their specific forms) and possible outputs. These parameters and the specification of data structures for computational representation have a significant impact on the characteristics and performance of the optimization process. Previous research in applications [8, 17, 1] has largely consisted and relied upon experimental analysis and intutition for designs and parameter settings. This paper takes a theoretical approach to the analysis of a specific design of a fuzzy rule base optimization system that has been used in a range of successful applications [6, 7, 11, 13]; we utilize the symetries that are inherent in the formulation to gain insight into the optimization. This leads to an interesting alternate viewpoint of the problem that may in turn lead to new approaches.

In particular, our formal definition and framework for the fuzzy rule base turns the optimization problem into a smooth problem that can be analyzed analytically. This analysis reduces the problem to a system of quadratic equations whose solution space has the surprising property that it generically contains a whole line. It should be possible to utilize this fact in the construction of fast and efficient solvers, which will be an important application of this research. The approach in this paper builds on experimental research presented in [7, 6], but it should be noted that a number of other mechanisms have been proposed for encoding fuzzy rules [8].

The methods we consider could be used in an evaluation process where the error is minimized with respect to fitting rule bases to some training data — in the context of the above example this would allow a system to learn rules with an output that is directly calculated from the data. For example a rule base evaluated in this way could be used to forecast the probability that a stock has positive price movement [10, 12] in some future time period. A rule in such a rule base could look like: If *Price to Earnings Ratio* is *Extremely Low* and *Double Moving Average Buy* is *Very High* then *probability of positive price movement* is 0.75. In this case the training data set would be some historical stock market data similar to that used in [6, 7].

The structure of this paper is as follows: Section 2 contains the formal definitions for the analysis presented in Section 3. Section 4 concludes the paper.

2 Approach

In this section we introduce the formulation of the models used in the analysis, including the rule base solution representation, the rule base interpretation method and the evaluation function.

2.1 Rule Base Solution Representation and Interpretation

Let us introduce some precise definitions of what is meant by the rule base solution representation. First of all, we are given L linguistic variables $\{A^1, ..., A^L\}$. Each linguistic variable A^i has M_i linguistic descriptions $\{A^i_1, ..., A^i_{M_i}\}$ that are represented by triangular membership functions μ^i_j, $j = 1, ..., M_i$. A fuzzy rule has the form

$$\text{If } A^{i_1} \text{ is } A^{i_1}_{j_1} \text{ and } A^{i_2} \text{ is } A^{i_2}_{j_2} \text{ and } \cdots \text{ and } A^{i_k} \text{ is } A^{i_k}_{j_k} \text{ then } o, \tag{1}$$

where $i_1, ...i_k \in \{1, ..., L\}$, $j_k \in \{1, ..., M_{i_k}\}$ and $o \in [0, 1]$.

A rule base is a set of several rules. Let us assume that we are given a rule base consisting of n rules:

$$\begin{array}{llll}
\text{If } A^{i^1_1} \text{ is } A^{i^1_1}_{j^1_1} \text{ and } A^{i^1_2} \text{ is } A^{i^1_2}_{j^1_2} & \text{and } \cdots \text{ and } A^{i^1_{k_1}} \text{ is } A^{i^1_{k_1}}_{j^1_{k_1}} & \text{then} & o^1 \\
\text{If } A^{i^2_1} \text{ is } A^{i^2_1}_{j^2_1} \text{ and } A^{i^2_2} \text{ is } A^{i^2_2}_{j^2_2} & \text{and } \cdots \text{ and } A^{i^2_{k_2}} \text{ is } A^{i^2_{k_2}}_{j^2_{k_2}} & \text{then} & o^2 \\
& \vdots & & \vdots \\
\text{If } A^{i^n_1} \text{ is } A^{i^n_1}_{j^n_1} \text{ and } A^{i^n_2} \text{ is } A^{i^n_2}_{j^n_2} & \text{and } \cdots \text{ and } A^{i^n_{k_n}} \text{ is } A^{i^n_{k_n}}_{j^n_{k_n}} & \text{then} & o^n,
\end{array}$$

where $i^m_l \in \{1, ..., L\}$ and $j^m_l \in \{1, ..., M_{i^m_l}\}$. Given a vector $x \in \mathbb{R}^L$ of observed values, whose components are values for the linguistic variables $A^1, ..., A^L$, we can evaluate the rule base as follows: the function ρ describes the way the rule base interprets data observations x to produce a single output value. This value has an application specific meaning and can be taken to be a real number (usually normalized to lie between zero and one). More precisely, ρ is defined as follows:

$$\rho : \mathbb{R}^L \to \mathbb{R}$$

$$x = \begin{pmatrix} x^1 \\ x^2 \\ \vdots \\ x^L \end{pmatrix} \mapsto \frac{\sum_{m=1}^n o^m \prod_{l=1}^{k_m} \mu^{i^m_l}_{j^m_l}(x^{i^m_l})}{\sum_{m=1}^n o^m}.$$

2.2 Evaluation Function

We consider an evaluation function (to minimize) that measures the error when training a rule base to fit a given data set. This *training data* consists of a set $\{x_i, y_i\}_{i=1...N}$, where each

$$x_i = \begin{pmatrix} x_i^1 \\ x_i^2 \\ \vdots \\ x_i^L \end{pmatrix}$$

is a vector that has as many components as there are linguistic variables, i.e. $x_i \in \mathbb{R}^L \ \forall\ i = 1, ..., N$, and each y_i is a real number, i.e. $y_i \in \mathbb{R} \ \forall\ i = 1, ..., N$. Then the evaluation function has the form

$$\epsilon = \sum_{i=1}^{N} (\rho(x_i) - y_i)^2 \tag{2}$$

$$= \sum_{i=1}^{N} \left(\frac{\sum_{j=1}^{n} a_{ij} o^j}{\sum_{j=1}^{n} o^j} - y_i \right)^2 , \tag{3}$$

where

$$a_{sm} = \prod_{l=1}^{k_m} \mu_{j_l^m}^{i_l^m} (x_s^{i_l^m}).$$

Our aim is to optimize the rules base in such a way that the evaluation function ϵ becomes minimal. This involves two separate problems. Firstly, the form of the membership functions μ_j^i may be varied to obtain a better result. Secondly, the rule base may be varied by choosing different rules or by varying the weights o^i. In this paper we will concentrate on the second problem, taking the form of the membership functions to be fixed. For example, we can standardize the number of membership functions for each linguistic variable A^i to be $M_i = 2n_i - 1$ and define

$$\mu_j^i = \begin{cases} 0 & : \ x \leq \frac{j-1}{2n_i} \\ 2n_i x + 1 - j & : \ x \in \left[\frac{j-1}{2n_i}, \frac{j}{2n_i}\right] \\ -2n_i x + 1 + j & : \ x \in \left[\frac{j}{2n_i}, \frac{j+1}{2n_i}\right] \\ 0 & : \ x \geq \frac{j+1}{2n_i} \end{cases}$$

for $j = 1, ..., 2n_i - 1 = M_i$. These functions are shown in Figure 1.

Moreover, we can consider the number n of rules to be fixed by either working with a specific number of rules that we want to consider, or by taking n to be the number of all possible rules (this number will be enormous, but each rule whose optimal weight is zero, or sufficiently close to zero can just be ignored and

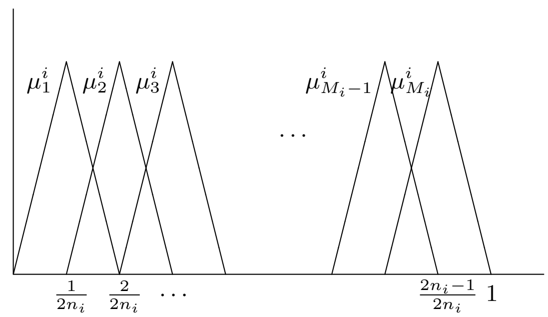

Fig. 1. Membership Functions

most weights will be of that form), depending on the application. The resulting optimization problem will be considered in 3.2.

3 Analysis

This section contains the detailed analysis of the problem described in Section 2. We firstly determine the maximum possible number of rules and then consider the optimization problem for the evaluation function. As a result, we are able to reduce the optimization problem to a system of equations (6), that has the remarkable property that it allows (generically) a one-dimensional solution space. This is the content of Theorem 1.

3.1 Search Space

The search space is the set of all potential rule base solutions. Let us first of all compute the maximum number of rules $n_{\max}$ that we can have. Each rule can be written in the form

$$\text{If } A^1 \text{ is } A^1_{j_1} \text{ and } A^2 \text{ is } A^2_{j_2} \text{ and } \cdots \text{ and } A^L \text{ is } A^L_{j_L} \text{ then } o,$$

where in this case $j_i \in \{0, 1, ..., M_i\}$ and $j_i = 0$ implies that the linguistic variable A^i does not appear in the rule. Then we have

$$n_{\max} = (M_1 + 1) \times (M_2 + 1) \times \cdots \times (M_L + 1) - 1.$$

Note that we have subtracted 1 to exclude the empty rule. If we include the possible choices of weights o^i with discretization $o^i \in \{0, \frac{1}{d}, ..., 1\}$, then we have a system of

$$(d + 1)^{n_{\max}}$$

possible rule bases.

3.2 Optimization Problem

In this subsection we will treat the optimization problem described in 2.2. We have to take the training data $\{x_i, y_i\}_{i=1\ldots N}$ and the various membership functions μ_j^i as given, so we can treat the various a_{ij} as constants and simplify

$$
\begin{aligned}
\epsilon(o) &= \sum_{i=1}^{N} \left(\frac{\sum_{j=1}^{n} a_{ij} o^j}{\sum_{j=1}^{n} o^j} - y_i \right)^2 \\
&= \sum_{i=1}^{N} \left(\frac{\sum_{j=1}^{n} (a_{ij} - y_i)^2 o^j o^j + 2 \sum_{j<k} (a_{ij} - y_i)(a_{ik} - y_i) o^j o^k}{\left(\sum_{j=1}^{n} o^j\right)^2} \right) \\
&= \frac{\sum_{j=1}^{n} A_{jj} o^j o^j + 2 \sum_{j<k} A_{jk} o^j o^k}{\left(\sum_{j=1}^{n} o^j\right)^2} \\
&\quad \text{with } \; A_{jk} = \sum_{i=1}^{N} (a_{ij} - y_i)(a_{ik} - y_i) \\
&= \frac{\sum_{j=1}^{n} \sum_{k=1}^{n} A_{jk} o^j o^k}{\left(\sum_{j=1}^{n} o^j\right)^2}.
\end{aligned}
$$

We want to find weights o^i such that this expression becomes minimal. In our formulation this requirement is smooth in the o^i, so we can compute the partial derivatives of the evaluation function with respect to the weights. At a minimal point $o_{\min} \in \mathbb{R}^n$, we must have

$$\frac{\partial \epsilon}{\partial o^1}(o_{\min}) = 0, \frac{\partial \epsilon}{\partial o^2}(o_{\min}) = 0, ..., \frac{\partial \epsilon}{\partial o^n}(o_{\min}) = 0.$$

It will turn out that this requirement is equivalent to a system of quadratic equations. So let us compute

$$
\begin{aligned}
\frac{\partial \epsilon}{\partial o^q}(o) &= 2 \frac{\left(\sum_{i=1}^{n} A_{iq} o^i\right) \left(\sum_{k=1}^{n} o^k\right) - \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} o^i o^j}{\left(\sum_{i=1}^{n} o^i\right)^3} \qquad (4) \\
&= \frac{2}{\left(\sum_{i=1}^{n} o^i\right)^3} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} (A_{iq} - A_{ij}) o^i o^j \right). \qquad (5)
\end{aligned}
$$

If we can simultaneously solve these n equations

$$\frac{\partial \epsilon}{\partial o^1}(o) = 0, \frac{\partial \epsilon}{\partial o^2}(o) = 0, ..., \frac{\partial \epsilon}{\partial o^n}(o) = 0,$$

then we have found a local extrema. For only two rules, for example, we obtain

$$
\begin{aligned}
\frac{\partial \epsilon}{\partial o^1}(o) &= \frac{2o^2}{(o^1 + o^2)^3} \left((A_{11} - A_{12}) o^1 + (A_{21} - A_{22}) o^2 \right) \\
\frac{\partial \epsilon}{\partial o^2}(o) &= \frac{2o^1}{(o^1 + o^2)^3} \left((A_{12} - A_{11}) o^1 + (A_{22} - A_{21}) o^2 \right)
\end{aligned}
$$

Therefore, if we assume that $o^1 \neq 0$ or $o^2 \neq 0$, then the optimal solution is

$$o^1 = \frac{A_{22} - A_{21}}{A_{11} - A_{12}} o^2.$$

This is a whole line that intersects zero in $\mathbb{R}^2$. This phenomena can be seen clearly in the following picture:

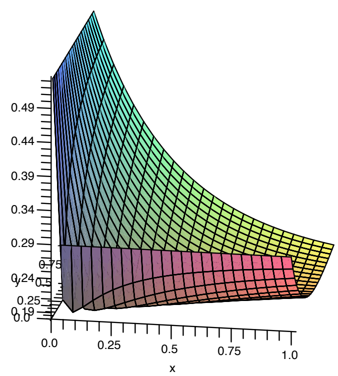

Fig. 2. Evaluation function for two rules

More than two rules If we have more than two rules, then the conditions become

$$\frac{\partial \epsilon}{\partial o^q} = 0 \Leftrightarrow \left(\sum_{i=1}^{n} \sum_{j=1}^{n} (A_{iq} - A_{ij}) o^i o^j \right) = 0, \; q = 1, ..., n. \tag{6}$$

Theorem 1. *Generically, there exists a one-parameter family of solutions to the system (6). Hence the space of extremal points for ϵ is a line in $\mathbb{R}^n$ that passes through zero.*

Proof. We will show that the n equations (6) are dependent, i.e. that we only need to solve $n-1$ of these equations and the n-th equation then follows automatically. For this purpose, we rewrite the system

$$\left(\sum_{i=1}^{n}\sum_{j=1}^{n}(A_{iq}-A_{ij})o^i o^j\right) = \sum_{\substack{j=1\\ j\neq q}}^{n} o^j \underbrace{\left((A_{qq}-A_{qj})o^q + (A_{jq}-A_{jj})o^j\right)}_{B_{qj}}$$
$$+\sum_{\substack{j=1\\ j\neq q}}^{n} \sum_{\substack{i=1\\ i\notin\{q,j\}}}^{n} \left((A_{iq}-A_{ij})o^i o^j\right).$$

Note that

$$B_{qj} = -B_{jq}.$$

Denote the q-th equation by E_q. Using the equality above, we compute

$$\sum_{k=1}^{n} o^k E_k = \sum_{k=1}^{n}\sum_{\substack{j=1\\ j\neq q}}^{n} \underbrace{B_{kj}o^k o^j}_{=0}$$
$$+\sum_{k=1}^{n}\sum_{\substack{j=1\\ j\neq q}}^{n}\sum_{\substack{i=1\\ i\notin\{q,j\}}}^{n}\left(\underbrace{(A_{ik}-A_{ij})}_{C_{ijk}} o^i o^j o^k\right)$$
$$= 0.$$

The last term vanishes due to the fact that the tensor C_{ijk} is symmetric in the index pair (i,j), symmetric in the index pair (i,k) and skew (i.e. anti-symmetric) in the index pair (j,k). Such a tensor has to vanish identically. It is hence sufficient to solve (6) just for $(n-1)$ equations, the last equation is automatically satisfied. □

Remark Given realistic training data, it is obvious that the extremal points that lie in $[0,1]^n$ will be minimal.

4 Conclusions and Future Work

We have successfully reduced the problem of finding optimal weights o^i for a rule base (given an arbitrary set of training data points) to a system of n equations for n unknowns, where n is the number of rules. Moreover, we have shown that the space of extremal points for the evaluation function is a line through the origin in $\mathbb{R}^n$. Hence a genetic algorithms will be able to find an optimal solution in $[0,1]^n$ using well-established and fast methods [3, 4]. The reason for this, somewhat surprising, result lies in the specific form of our rule base formulation: not the values of the weights themselves are important, but the relationship that they have with respect to each other. Mathematically speaking, the optimal solution o is really an element of $(n-1)$-dimensional projective space $\mathbb{RP}^{n-1}$, rather that an element of $\mathbb{R}^n$.

As a result, it is possible to use the analysis in this paper to design an optimization process in which combinations of rules consisting of the *if* parts are selected and then evaluated using a fast algorithm to find the optimal *then* parts (output weights) to produce a rule base. This would be beneficial in a range of applications as mentioned in the introduction. For example, reducing the size of the search space by removing the assignment of output weights from the general rule base search problem; or by fixing an initial structure for the rules (using knowledge from the application domain or because of specific application requirements) that feasible solutions should contain and then redefining the search objective to extend this set rather than using a free search — this is for instance a very useful feature in financial applications [7]. As a part of this further research, we will also examine, combinatorially and empirically, algorithms and genotype representations that utilize the reduction in complexity that arises from the analysis in this paper.

References

1. F. Allen and R. Karjalainen. Using genetic algorithms to find technical trading rules. *Journal of Financial Economics*, 51:245–271, 1999.
2. A. Bai. Determining fuzzy membership functions with tabu search – an application to control. *Fuzzy Sets and Systems*, 139(1):209–225, 2003.
3. N. Courtois, L. Goubin, W. Meier, and J.-D. Tacier. Solving underdefined systems of multivariate quadratic equations. In *PKC '02: Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptosystems*, pages 211–227. Springer-Verlag, 2002.
4. L. D. (ed). Handbook of genetic algorithms. 1991.
5. A. Esmin and G. Lambert-Torres. Evolutionary computation based fuzzy membership functions optimization. In *Systems, Man and Cybernetics ISIC IEEE International Conference on*, pages 823–828. IEEE, 2007.
6. A. Ghandar, Z. Michalewicz, M. Schmidt, T.-D. To, and R. Zurbruegg. A computational intelligence portfolio construction system for equity market trading. In *CEC 2007: Proceedings of the 2007 IEEE Congress on Evolutionary Computation.* IEEE Press, 2007.
7. A. Ghandar, Z. Michalewicz, M. Schmidt, T.-D. To, and R. Zurbruegg. Computational intelligence for evolving trading rules. *To appear in IEEE Transactions On Evolutionary Computation, Special Issue on Evolutionary Computation for Finance and Economics*, 2008.
8. J. Gomez. Evolution of fuzzy rule based classifiers. In *GECCO2004 Proceedings of the Genetic and Evolutionary Computation Conference*, volume 3102, pages 1150–1161. Springer-Verlag, 2004.
9. N. R. Guo, T.-H. S. Li, and C.-L. Kuo. Design of hierarchical fuzzy model for classification problem using gas. *Comput. Ind. Eng.*, 50(1):90–104, 2006.
10. N. Jegadeesh and S. Titman. Profitability of momentum strategies: An evaluation of alternative explanations. 56(2):699–720, 2001.

11. R. Johnson, M. Melich, Z. Michalewicz, and M. Schmidt. Coevolutionary optimization of fuzzy logic intelligence for strategic decision support. *IEEE Transactions on Evolutionary Computation*, 9(6):682–694, 2005.
12. M. Lo, A.W., H., and J. Wang. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. 55(4):1705–1765, 2000.
13. Z. Michalewicz, M. Schmidt, M. Michalewicz, and C. Chiriac. A decision-support system based on computational intelligence: A case study. *IEEE Intelligent Systems*, 20(4):44–49, 2005.
14. M. Mucientes, L. Moreno, A. Bugarin, and S. Barro. Evolutionary learning of a fuzzy controller for wall-following behavior in mobile robotics. *Soft Comput.*, 10(10):881–889, 2006.
15. B. Oduor and A. Anev. Evolving fuzzy systems. *J. Comput. Small Coll.*, 19(5):356–357, 2004.
16. W. Pedrycz. Why triangular membership functions? *Fuzzy Sets Syst.*, 64(1):21–30, 1994.
17. E. H. Ruspini, P. P. Bonissone, and W. P. (eds.). Handbook of fuzzy computation. 1998.
18. Y. Shi, R. Eberhart, and Y. Chen. Implementation of evolutionary fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 7(2):109–119, 1999.
19. H. Surmann and M. Maniadakis. Learning feed-forward and recurrent fuzzy systems: a genetic approach. *J. Syst. Archit.*, 47(7):649–662, 2001.
20. C.-J. Wu and G.-Y. Liu. A genetic approach for simultaneous design of membership functions and fuzzy control rules. *J. Intell. Robotics Syst.*, 28(3):195–211, 2000.