Techniques for Developing Large Scale Fuzzy Logic Systems

Jon C. Ervin
Apogee Research Group
Los Osos, CA 93402, USA
apogee@alumni.calpoly.edu

Sema E. Alptekin
California Polytechnic State University
San Luis Obispo, CA 93407, USA
salpteki@calpoly.edu

Abstract

In this paper, we describe various techniques used to make Intuitionistic Fuzzy Logic Systems amenable to operating on applications with large numbers of inputs. A rule reduction technique known as Combs method is combined with an automated tuning process based on Particle Swarm Optimization. A second stage of tuning on rule weights results in improved performance and further reduction in the size of the rule-base. The entire process has been developed to operate within the Matlab software environment. The technique is tested against the Wisconsin Breast Cancer Database. The use of these tools shows great promise in significantly expanding the range and complexity of problems that can be addressed using Atanassov's Intuitionistic Fuzzy Logic.

Keywords: Atanassov's Intuitionistic Fuzzy Logic, Particle Swarm Optimization, Combs Method.

1 Introduction

The decision making paradigm known as Fuzzy Logic has been implemented in a wide range of applications since it was first introduced by Dr. Lotfi Zadeh in his seminal paper published in 1965 [17]. A number of researchers have since made major contributions in expanding and adapting Fuzzy Logic beyond its initial conception. One major Achilles heel for Fuzzy Logic applications has been that in most current algorithms, memory requirements expand exponentially with linear growth in inputs. This property of the formulation can quickly overwhelm available computing resources in even relatively modest sized applications. One promising extension, Intuitionistic Fuzzy Logic (IFL), can further exacerbate memory problems by effectively doubling the number of input membership functions. Atanassov [1] introduced the notion of pairing non-membership functions in IFL with the usual membership functions of Fuzzy Logic.

A number of researchers have developed techniques to mitigate the problem "exponential rule explosion" or "curse of dimensionality" through various means [8, 10, and 14], however, these methods generally tend to be complex and limited to special cases. One promising exception is a technique developed by William Combs that changes the problem from one of an exponential dependence to a linear dependence on the number of inputs [4]. The use of Combs method also simplifies rule-base generation and makes it easier to automate tuning of Fuzzy systems.

Another issue that must be addressed when implementing a Fuzzy Logic algorithm for a specific application is constructing membership functions for each input that provide adequate system performance. A common approach taken is to interview experts in the particular field and use their feedback to build the fuzzy membership functions. This process can be cumbersome and/or completely impractical for large problems, leading to the need to implement some sort of automated membership function construction and optimization process.

A number of methods for automating this process have been published in the literature. A common approach has been to somehow construct membership functions and then use a population based optimization method such as the Genetic Algorithm to tune them. We chose to follow a variation of this approach by constructing the membership functions and using a method known as Particle Swarm Optimization (PSO) for tuning. The PSO algorithm, first developed by Eberhart and Kennedy [6], was inspired by the coordinated group behaviors of animals such as flocks of birds in flight. One important advantage of this technique is the simple formulation and implementation of the underlying equations.

The main focus of our research has been to combine several techniques to simplify the tuning process and make it practical to address problems of much greater size and complexity [7]. Recently, we have built on previous research to further improve the optimization process. We believe our efforts have brought us closer to realizing an efficient, automated process to generate Fuzzy Logic systems capable of handling large and complex problems.

The following section provides a brief background of Combs method and the database used to verify our approach. The next section describes the PSO optimization process. Following that we provide some of our results and end with a section on our conclusions.

2 Combs URC Method

William Combs refers to his method as the union rule configuration (URC) versus what he calls the intersection rule configuration (IRC) of the traditional fuzzy rule-base construct [4]. The main difference between the URC and IRC is that every rule in the rule-base of the URC is required to have only one antecedent for every consequent. Initially, this may sound counter-intuitive as a means of reducing the number of rules, however by imposing this restriction it means that each membership function of all the input variables is used only once in the antecedent of a rule in the rule base. Each of these rules are joined by a logical OR in the rule-base, hence the designation union rule configuration.

Combs and his various co-authors show that the entire problem space can be accessed by implementing the URC. A spirited debate can be found in the literature discussing the validity of the claims made for the URC and the interested reader is referred to the references [3, 5, 12, and 15] for detail on this topic. Our own experience to date has been that the URC performs as well or better than the IRC formulation.

We used Combs method to apply Atanassov's Intuitionistic Fuzzy Logic to the well known Wisconsin Breast Cancer Database (WBCD) [16]. An initial set of four IFL membership and non-membership functions were developed for each of the nine input variables of the WBCD. The 683 unique samples in this database associate various diagnostic tests as real-valued inputs with a binary output of benign/malignant for a suspect tissue mass. The WBCD was chosen as a test case because it provides a relatively large number of inputs, allowing us to demonstrate the rule-base reduction advantage of the URC method. The entire rule-base consisted of 36 rules (9 inputs x 4 membership/non-membership functions) versus the 262,144 (4^9) maximum number of rules that could be used in the IRC method.

Figure 1 shows the distribution of the values of one of nine WBCD input variables with respect to benign/malignancy of the tissue mass. On this chart a diagnosis of benign is marked as an "x" and a diagnosis of malignant is marked with an "o". The distribution shows a strong correspondence between low values and a diagnosis of benign, however there are a number of exceptions to this generalization. This distribution is characteristic to a greater or lesser degree in all nine of the input variables.

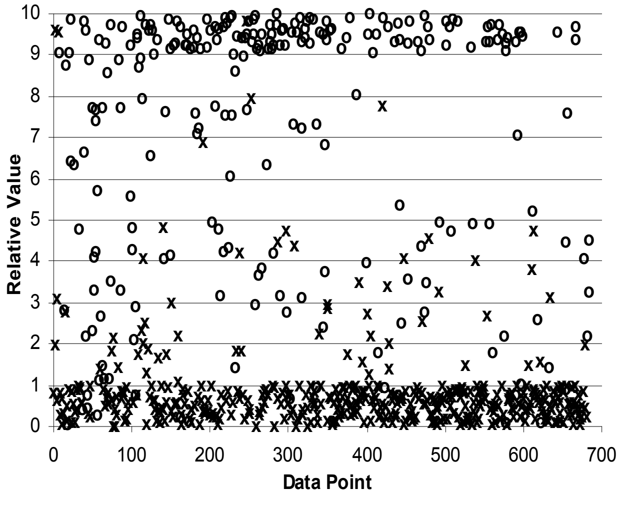


Figure 1: Distribution of Input "Bare Nuclei"

3 Fuzzy Classification Model

In the spirit of simplifying the development and tuning processes, the entire optimization algorithm was created, debugged and executed within the Matlab software suite [11]. The intuitionistic fuzzy classifier was created in the Matlab Fuzzy Logic Toolbox. A set of customized Matlab m-files provide overall control of the optimization process. The m-files make calls to the PSO and IFL modules, with the program terminating after a user-defined number of iterations have been completed.

The Fuzzy Logic Toolbox provides a number of general default values for the user to choose from in building the fuzzy classification model. For instance, the software allows the user to choose between two methods, minimum or product, to fill in for the "AND" function in the rule antecedent. In our case we chose to employ the product method. Other standard parameters selected were; weighted average for defuzzification, the Sugeno-type inference method, and rectangular membership functions.

The output values from the IFL module were continuous over the interval [0 1]. Any output number that was .5 or less was assigned a diagnosis value of benign, and conversely a number greater than .5 was assigned to be malignant. It was considered desirable that in the final optimized fuzzy classifier not only should the output provide as many correct diagnoses as possible, but also that the raw output numbers should be as close as possible to the extremes of the [0 1] interval thus reducing ambiguity in each diagnosis.

4 Optimizing with PSO

Since the rule-base is pre-determined in our approach to the URC, the initial focus of the optimization process was on the definition of the (non)membership functions. In this most recent work, further rule-base reduction was achieved through the optimization of weighting factors for each individual rule.

Each input variable was assigned 2 membership and 2 non-membership trapezoidal shaped functions. These functions were given values of low/high and (not high)/(not low) respectively. Two of the four points that determined each trapezoid were anchored to one side of the allowable input range depending on whether the function described a predominately low or high input value. This action left two points of each membership function to float freely over the input range and it was these two values that were subjected to the optimization process.

The total number of optimization variables was: (9 input variables) x (2 points per function) x (4 functions per input variable) = 72 variables.

Figure 2 shows an example of the (non)membership function for one input variable with the 8 points subjected to optimization circled. Here we abbreviate "membership and non membership" with "(non)membership".

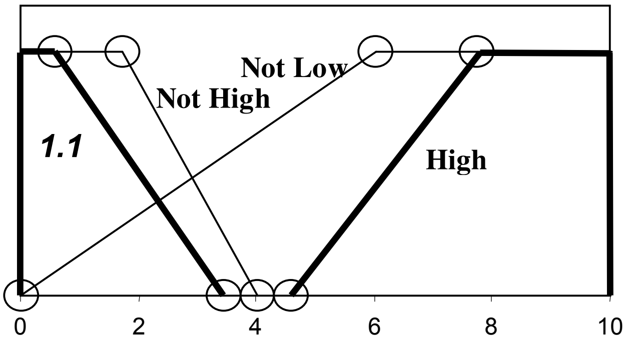


Figure 2: Four (non)Membership Functions with Eight Circled Optimization Variables

Particle Swarm Optimization (PSO), like the Genetic Algorithm is a population-based optimization method inspired by biological phenomena. In the case of PSO the inspiration comes from flocking behaviors of birds or schooling of fish. An optimization run is initialized by dispersing a population of solutions at random throughout the N-dimensional problem space. A new position for each of the solutions or "particles" is then calculated based on the generating equations:

$$V_{id}^{+1} = V_{id} + c_1 r_1 (X_{id}^{best} - X_{id}) \quad (1)$$

$$+ c_2 r_2 (G_d^{best} - X_{id})$$

$$X_{id}^{+1} = X_{id} + V_{id}^{+1} \quad (2)$$

$i = 1,\ldots M$ Population
$d = 1,\ldots N$ Dimensions

where X_{id} is the particle position vector and V_{id} is an associated "velocity" vector. The predetermined constant coefficient c_1 is often referred to as the cognitive parameter and c_2 as the social parameter in the velocity vector. The random numbers r_1 and r_2 are selected from a uniform distribution on the interval [0,1] and X_{id}^{best} and G_d^{best} are the previous personal best position for each individual particle and the global best position of the population, respectively.

An excellent add-on to the Matlab software suite, the PSO Toolbox, is distributed free for use on the internet [2]. We modified the source code of this package to interface with the Fuzzy Logic Toolbox that is also an add-on to Matlab software suite [9]. A flow diagram of the membership function optimization process is shown in Figure 3.

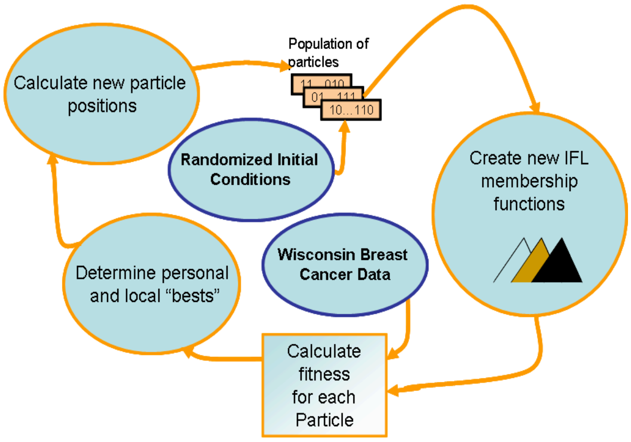
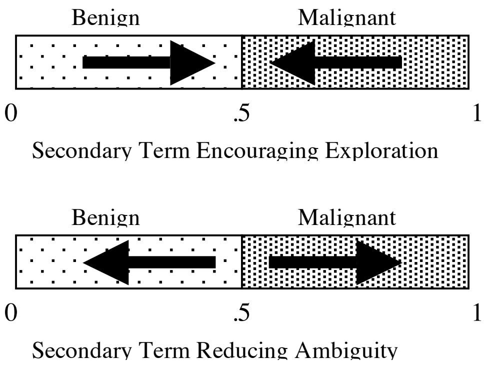

Population of particles
11...010
01...111
10...110
Calculate new particle positions
Randomized Initial Conditions
Create new IFL membership functions
Determine personal and local “bests”
Wisconsin Breast Cancer Data
Calculate fitness for each Particle

Figure 3: IFL Membership Function Optimization

Like the Genetic Algorithm, PSO requires a fitness function to rank the worthiness of candidate solutions. The output values from the IFL system ranged continuously over the interval [0, 1], any value below 0.5 was assigned to be a diagnosis of benign and any value above 0.5 was assigned as malignant. Initially, there were two terms in the fitness function that we designed as a minimization problem.

The first term was dominant and served to ensure that best fitness was assigned to the candidate IFL system that gave the fewest number of incorrect diagnoses. A secondary term in the fitness function served to minimize the ambiguity in correct diagnoses by penalizing values that came close to a value of 0.5. This secondary term served as a tie breaker between multiple candidates that had the same number of misdiagnoses by giving preference to the candidate that maximized the root mean squared distance from the value of 0.5 for each set of input values. The goal for this secondary term was to reduce “ambiguity” or sensitivity of the output to small variations in the input values.

In a later refinement the secondary ambiguity term in the fitness function was modified in such a way that for extended periods during the optimization process the action of this term was reversed to encourage convergence ***toward*** a value of 0.5. This was done in order to encourage exploration by the search mechanism by requiring a minimum amount of perturbation to achieve a possible improvement in the number of misdiagnoses. In these cases a large number of epochs were reserved at the end of an optimization run with the secondary fitness term switched to driving convergence away from a value of 0.5 and reducing ambiguity in the final result. Figure 4 illustrates this refinement in the secondary term of the fitness function.



Secondary Term Encouraging Exploration

Secondary Term Reducing Ambiguity

Figure 4: Effect of the Secondary Fitness Term on Convergence

An additional step was then implemented that carries on from the process described above. In this process the IFL system with optimized membership functions was subjected to a re-optimization, this time of weighting factors for each rule in the rule-base. This re-optimization process was again performed using the PSO algorithm, only this time a weighting factor for each of the 36 rules in the rule-base were the assigned optimization input variables. This second optimization process followed the same flow as that shown in Figure 3 with rule weight creation replacing the membership function creation block.

In these rule optimization runs a third term was added to the fitness function that served to minimize the total number of rules utilized by the optimized fuzzy classifier. The interest for this feature was to reduce the number of rules on top of the gains contributed by using the URC method, resulting in even faster computer run times. This would be an important feature for very large fuzzy classification problems.

At first, it seemed likely that this third term would conflict with the ambiguity role of the second term. As it turned out however, incorporating the rule reduction term had only a slight negative impact on the secondary ambiguity goal.

The improvement in the fitness function during an optimization run followed an exponential decay, showing rapid improvement early on and much slower gains with increasing numbers of epochs. The PSO method provided convergence in a reasonable amount of time on a relatively modest computing platform. The method also was easy to formulate and code into a working program in a short amount of time.

5 Results

The optimization process successfully produced an Atanassov's Intuitionistic Fuzzy System that provided results similar to that found by other authors [13]. The best outcome produced a system that gives a correct diagnosis 98.975% of the time or 7 misdiagnoses out of 683 cases. In the second phase of the tuning process, rule weight tuning produced further improvement gains in the secondary fitness term. In addition, 16 of the 36 rules were eliminated completely during this process. This resulted in a 44% reduction in the rule-base and membership function definitions, thus decreasing system memory requirements and increasing execution speed. It should be noted that these improvements occurred on top of the already reduced rule base size achieved through using the Combs method.

A separate optimization run was made using Fuzzy Logic membership functions only. The best performance that could be achieved was a system that gave 9 misdiagnoses out of 683 cases or an accuracy of 98.68%. Therefore the membership function only case produced 2 additional or 28.57% more misdiagnoses. In a large population this improvement might result in a significant number of lives saved from misdiagnoses. A typical optimization run with a population of 120 particles running for 60,000 epochs would finish in 36 hours on a Dell Latitude D600 laptop.

6 Conclusions

An Atanassov's intuitionistic fuzzy system was optimized to produce a correct breast cancer diagnosis with an accuracy that rivaled that of the best systems to date. The IFL employed Combs URC rule construction methodology to limit rule-base growth to a linear relationship with increasing numbers of inputs and membership functions. The optimization process proceeded in two stages. Using Particle Swarm Optimization, membership functions for the IFL system were first optimized to reduce misdiagnoses to 7 out of 683 cases. In the second phase of optimization PSO was again used to tune rule weights, resulting in a better tuned system with a reduced rule-base.

The entire process was developed and executed within the Matlab software suite. The combination of tools used provided a relatively automated process that required less memory and faster running times than would normally be expected for a large number of inputs. The resulting IFL system with (non)membership functions performed better than a similarly optimized standard Fuzzy system with membership functions only.

References

[1] Atanassov K., *Intuitionistic Fuzzy Sets*, Springer-Verlag, Heidelberg, 1999.

[2] Birge, B., Particle Swarm Optimization Toolbox, url: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=7506&ref=rssfeed&id=mostDownloadedFiles, last accessed on March 31, 2008.

[3] Combs, W.E. and J.E. Andrews, "Combinatorial Rule Explosion Eliminated by a Fuzzy Rule Configuration", IEEE Transactions on Fuzzy Systems, Vol. 6, No. 1, pp. 1-11, (1998).

[4] Cox, E., The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems – Appendix A, Academic, Boston, MA, (1994).

[5] Dick, S. and A. Kandel, "A comment on Combinatorial Rule Explosion Eliminated by a Fuzzy Rule Configuration by William E. Combs and James Andrews", IEEE Transactions on Fuzzy Systems, Vol. 7, No. 4, (1999), pp: 475-477.

[6] Eberhart, R. and J. Kennedy, "A New Optimizer Using Particle Swarm Theory", Proc. of the Sixth Int. Symposium on Micro Machine and Human Science, Oct 1995, pp. 39-43.

[7] Ervin, Jon C., and Sema E. Alptekin, "Combs Method Used in an Intuitionistic Fuzzy Logic Application", Proceedings of WILF 2007 - International Workshop on Fuzzy Logic and Applications, Special Session on "Intuitionistic Fuzzy Sets: Recent Advances, July 2007, Italy.

[8] Guvenc, M.K. and K.M. Passino, "Avoiding Exponential Parameter Growth in Fuzzy Systems", IEEE Transactions on Fuzzy Systems, Vol. 9, No. 1, February 2001.

[9] Jang, J. S. and N. Gulley, Fuzzy Logic Toolbox User's Guide, version 1, The MathWorks, Natick Mass., 1997.

[10] Linkens, D. A. and H. O. Nyongesa, "A Hierarchical Multivariable Fuzzy Controller for learning with Genetic Algorithms", Int. J. Control, Vol. 63, No. 5, 1996, pp. 865-883.

[11] Matlab, url: http://www.mathworks.com/ Last accessed March 31, 2008.

[12] Mendel, J. M. and Q. Liang, "Comments on Combinatorial Rule Explosion Eliminated by a Fuzzy Rule Configuration", IEEE Transactions on Fuzzy Systems", Vol. 7, No. 3, 1999, pp: 369-373.

[13] Pena-Reyes, C. A. and M.Sipper, "Fuzzy CoCo: A Cooperative Coevolutionary Approach to Fuzzy Modeling", IEEE Transactions on Fuzzy Systems 9 (5) (2001), pp: 727-737.

[14] Raju, G.V.S. and J. Zhou, "Adaptive Hierarchical Fuzzy Controller", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 4, (1993), pp: 973-980.

[15] Weinschenk, J. J., W.E. Combs, and R. J. Marks II, "Avoidance of rule explosion by mapping fuzzy systems to a disjunctive rule configuration", 2003 International Conference on Fuzzy Systems (FUZZ-IEEE), St. Louis, (May 25-28, 2003).

[16] Wolberg, W. H., Wisconsin Breast Cancer Database, url: http://www.ailab.si/orange/datasets.asp?Inst=on&Atts=on&Class=on&Values=on&Description=on&sort=Data+Set
last accessed on Jan 26, 2007.

[17] Zadeh, L.A., "Fuzzy Sets", Inf. Control 8, pp: 338-353, 1965.