Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Optimizing Fuzzy Clinical Decision Support Rules Using Genetic Algorithms

Michael Krajnak and Joel Xue, *GE Healthcare*

Abstract—**In this paper, we present a technique for optimizing a fuzzy system using a genetic algorithm that works for patient status monitoring in the operating room. The genetic algorithm adjusts rule weights, outputs, and input membership functions to maximize the area under a receiver operator curve (ROC) for final classification. Compared to pre-optimization, the optimized fuzzy inference system increased ROC area from 0.68 to 0.77, which can be translated to an increase in specificity from 74% to 82%, at a fixed sensitivity of 58%.**

I. Introduction

Fuzzy logic (or fuzzy inference system - FIS) has been widely used in control and pattern recognition applications, since they directly attack the problems associated with the dealing with large numbers of complex inputs and are well suited for dealing with ambiguous data or situations. Optimizing fuzzy systems can be difficult because internal system interactions are complex, and difficult to visualize. Purely data driven techniques can be used to automatically construct expert systems, but require high fidelity training data, which is difficult obtain when clinical pathologies are ambiguous, exactly the situations where fuzzy expert systems are the most useful. A key advantage of using fuzzy logic in clinical decision support is that the rules can be programmed and easily understand by clinicians, unlike neural networks and other regression approaches, where the system behaves as a black box IV.

In our research we have been looking at the application of fuzzy inference to identifying clinical conditions that occur during anesthesia delivery in the operating room. For example, during anesthetic induction, surgical incision, or other periods where increased anesthetic affect is desired, agent levels are typically stepped up for some time period. If left on too long the increased agent can result in hemodynamic instability and an undesirable decrease in blood pressure and heart rate. In a study of over 4,000 patients, 9% experienced severe hypotension in the period of 0-10 minutes after induction IV. Intraoperative hypotension is generally associated with the post-induction phase of anesthesia and hypotension is a significant factor in long-term mortality and morbidity IV. An automated system that detects this condition could help reduce the occurrence and severity of periods of low blood pressure.

Monitoring of physiologic variables can help identify developing hemodynamic instability due to anesthesia overdose. When a patients blood pressure and heart rate are decreasing, and the anesthetic agent has been recently increased over 1 MAC, the likelihood of anesthetic overdose is high IV. This type of clinical expertise is represented in a fuzzy inference rule set by using fuzzy values to represent terms such as "increasing", or "low". The effort to systematically identify clinical rules should not be underestimated, but it is not the focus of this paper.

There are a number of technical steps that must take place before a fuzzy rule set can be evaluated by an inference system:

1. Input parameters must be identified that correspond to the clinical inputs, such as heart rate.
2. The clinical inputs must be converted from their normal "crisp" representation to a fuzzy number using membership functions to assign inputs to fuzzy values such as "decreasing", "stable", or "increasing".
3. Each rule is associated with a weight that can increase or decrease its influence on the outcome.
4. Each rule is mapped to a fuzzy output value, such as a "high" likelihood of anesthetic overdose.

Membership function parameters, rule weights, and outcomes, can all be viewed as independent parameters that represent tuning opportunities to adjust the fuzzy inference system to more closely model the physiological condition that we are trying to observe.

The correlation between different inference system parameters and the clinical output is not always straight forward. Complications due to system interactions make it very difficult to predict how changing any one parameter will affect the outcome.

II. Methods

Our goal is to walk briefly through the overall inference system construction process and then show how a genetic algorithm based optimizer can be used to improve it. This technique has been used to optimize inference systems with up to six input features, but the examples shown here have only three.

A. Problem Definition

Identifying clinical conditions of interest, the general pathology of each condition, and related inference rules was done through reviews of existing literature and interviews

Manuscript received April 3, 2006.

Joe Xue is with GE Healthcare Information Technology, Wauwautosa, WI 53226 USA (e-mail: joel.xue@med.ge.com).

Michael Krajnak is with GE Healthcare Information Technology, Milwaukee, WI 53201 USA (414-362-2495; e-mail: michael.krajnak@med.ge.com).

with expert clinicians. We identified the fuzzy variables that corresponded to clinical terms and fuzzy values for those variables. A diagram of the inference system optimization system is show in Figure 1. There are 3 layers in the system, plus a parameter turning path for system optimization during training phase. The first layer consists of 3 inputs: slope of arterial mean blood pressure (dBp/dt), slope of heart rate (dHr/dt), and agent level; the second layer is a rule engine, and the third layer is output and defuzzification. The optimization path is introduced in section D.

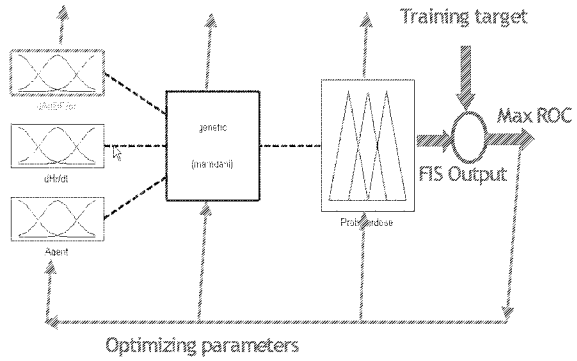

Figure 1. The diagram of the optimization system

B.Data Collection and Annotation

We then acquired data in conjunction with our clinical partners for each condition we were interested in. We are specifically interested in real time data that is available for clinical decision support such as monitored physiological parameters. Data from electronic medical records are used where possible to initialize and configure different algorithms, but the emphasis is on gathering real-time data.

Data is annotated retrospectively by two clinicians to identify clinically significant indicators of specific conditions. Data was not annotated as it was collected because we decided that the risk of distraction in the operating room was too great. By using two reviewers for each case we were able to minimize annotation variability that might otherwise exist in a retrospective review.

C.Initial FIS Construction and Analysis

Our initial FIS was constructed using verbal guidance from clinicians about approximate transition values for ranges such as low to moderate blood pressure, augmented by viewing input feature statistics, primarily simple averages, inside and outside of annotated episode regions.

We tried a number of simple improvements to increase performance. It quickly became apparent the benefit of any single improvement was difficult to predict due to the systems overall complexity and time consuming to evaluate.

In order to understand the limits of our ability to manually optimize the expert systems we benchmarked our expert systems against a neural network with standard back propagation and steepest descent training methods. The training results show that NN model could not form a good classification boundary due to ambiguous input parameters.

These results lead us to consider stochastic methods for searching for optimal solutions for our inference systems. Of the stochastic optimization methods we considered genetic algorithms were judged to be a best fit for our problem because of the large number of optimization dimensions and nonlinearity present in the problem.

D.Genetic Algorithm Construction

Genetic algorithms essentially perform a parallel stochastic search over all dimensions of the solution space and uses genetic operators rather than derivative based techniques to converge on a solution. The key components to a genetic algorithm are: encoding, fitness evaluation, selection, crossover, and mutation IV.

1) Encoding

In a genetic algorithm the free parameters are encoded as independent genes. All of the genes that describe a particular solution make up a genome. A simple encoding scheme will directly map a gene to a value. In our scheme we used simple integers to represent outcome and rule weight genes. Continuous values, like rule weights, were limited to discrete values. We did not encode the membership function parameters directly because of the complexity of constraining sets of membership functions to be normalized. Rather, we created a set of template membership functions that were indexed by a single gene and rescaled based on the expected minimum and maximum values for a given input feature. We also included genes that controlled the skew of the membership functions in order to cover a larger portion of the solution space. Our smallest genome encoded this way had 63 genes. The size increased dramatically as input parameters were added. Our largest genomes had 501 genes.

2) Fitness Evaluation

Our initial system used the area under the receiver operator curve (ROC) directly as the fitness, later we used the squared area under the curve to bias the selection operator towards more fit genomes. We considered using a fitness based on an error function derived from the difference between the system output and the expected values. However, our expected values were limited to 0 or 1, indicating the condition was not present or present. The behavior of any error function, especially in the transition regions, was sufficiently complex that we decided to use the ROC area metric instead. Also asking the clinicians to annotate the degree of clinical significance was ruled out by our clinicians because of the inherent ambiguity of the problem.

3) Selection

We used a selection operator which chose parent genomes proportionally with respect to their fitness. We also applied elitism, the keeping of a small percentage (0.10) of the highest fitness genomes from the current generation and moving them without change into the next generation.

4)Crossover

We used uniform crossover operator which uses a 50/50 probability of using either parent's gene to make up the child genome. We chose the uniform crossover operator in order to avoid biasing our search away from any portion of the solution space.

5)Mutation

From each new generation we choose 2% of the new genomes and changed the value of a single gene.

Some machine learning approaches for building inference systems include options that allow rules to be discovered during the learning process, for example NEFCON IV. We started with a full rule base so that inference system response surface completely covered the input space.

We constrained the input membership functions sets for a single variable to be normalized. There are neuro-fuzzy models IV, IV, that can tune input membership functions, but that do not keep the input membership function sets normalized. However, our process is limited to a small number of discreet possibilities and as such may be suboptimal. In previous experience with clinical inference systems we found that small changes in input membership functions had a minimal impact on the system.

We used a fixed set of output membership functions. For the clinical conditions of interest there was no clear bias between the fuzzy output values, and by keeping the output mapping static and tuning the system exclusively by updating rule weights, outputs, and input membership functions it was easier to compare different system inference rule sets.

In our current version of the optimizer most evolution runs used a fixed population of 100 genomes running over 160 to 200 generations. These numbers where arrived at using simple heuristics and observing several simple test runs.

We also restricted our current systems to Mandami style fuzzy inference systems. Systems with continuous linear output are often more amenable to other machine learning techniques IV, but we felt that models with fuzzy outputs captured the clinical thought process more appropriately so we choose to use the Madami model with fuzzy output values.

The progress the algorithm can be tracked by watching the evolution of the fitness values of the test populations by plotting the maximum, minimum, and median fitness for each generation. We evaluated the trajectory of the system in this way using several trial runs and selected reasonable values for the retention and mutation fractions and initial population size and number of generations per test run.

Once the initial system was operational we constructed a number of experiments by selecting appropriate configurations of the input features and running the optimizer for each system. Because of the stochastic nature of genetic algorithms each experiment was run three times to gauge the convergence of a particular configuration.

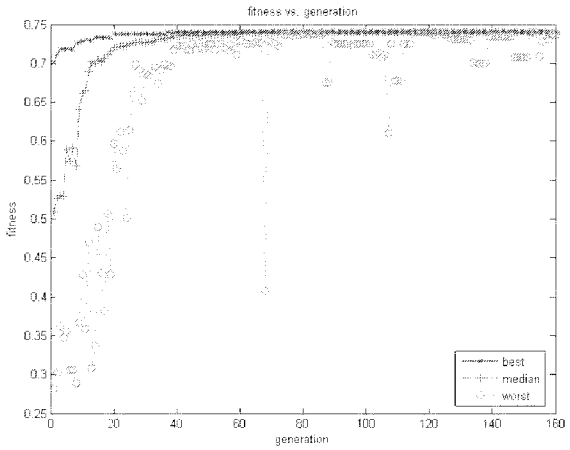

Figure 2. Fitness vs. generation.

III.Results

Here is one case from the set of optimization experiments we conducted that compares an early set of rules we constructed before the optimizer was created. We choose this earlier case primarily because once we started optimizing inference systems we manually created fewer rules sets from scratch, rather we used hybrid techniques and focused on manually improving rule sets produced by the optimizer.

This system has three fuzzy input variables each with three values. We created the initial membership functions as described in "Methods", and conducted additional trail and error experiments in order to tune the results. The area under the ROC curve for this expert system is 0.68.

For the genetic optimization process we used a data set of approximately 80 hours of high resolution parameter data. The initial genome population consisted of 100 individuals. The creation of new generations used uniform crossover algorithm with a retention fraction of 0.10 and a mutation fraction of 0.02. The optimizer was run for 160 generations. The best expert system had an area under the curve of 0.77. Figure 3 shows the ROC curves for the manual and optimized system.

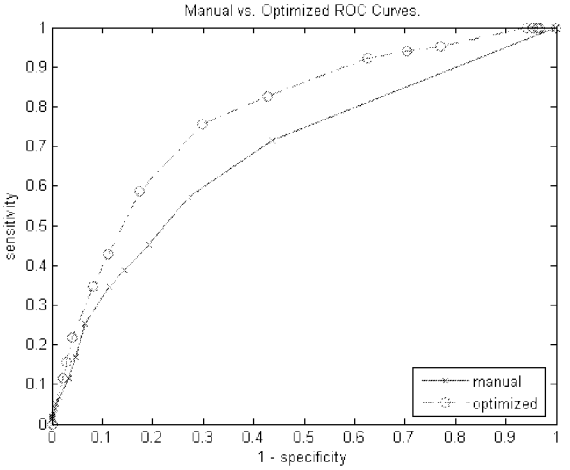

Figure 3. Manual and optimized ROC curves.

Figure 4 shows one of the input membership function (BP trend) before and after the optimization. The post-optimization input function shows a more smooth transition among its members, which is consistent with the nature of the ambiguity of this input variable.

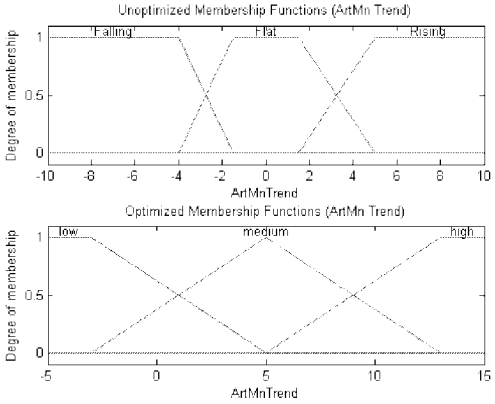

Figure 4. A input membership function before and after optimization.

The overall area under the curve is increased by 0.09. At 58% sensitivity the specificity is increased from 74% to 82%. Obtaining this result would have taken much longer using only manual optimization.

Another interesting feature of the ROC curves is that the optimized system maps more uniformly over the full range of sensitivities and specificities. This implies that the optimized inference system has better dynamic range if a variable threshold is used to adjust the system tradeoff between sensitivity and specificity.

IV. Discussion

We have used this procedure to optimize a wide variety of expert system rule sets. In every case the optimizer is able to improve over the original system.

Because of its relatively rapid turn around time the optimizer can be used to run through different combinations of inputs and enable differential comparison of options. In this way we can rapidly compare for example the impact of using heart rate from SpO2 measurements versus heart rate from ECG. If we attempted to make these comparisons with manually tuned inference systems the risk that the differences were due inappropriate choices of system parameters instead of actual performance would be relatively high. Other tradeoffs can also be explored this way, for example choices in input preprocessing algorithms.

When the optimizer begins a run, it makes no assumptions about the dependent variables, even if we have constructed rule sets manually before optimization. If we included a manually tuned set of rules early in the population, it is likely that it would quickly dominate the population and limit its genetic diversity and the corresponding search. This does not mean that we are trying to discover or mine rules from scratch from the data. Because the conditions we are trying to detect are difficult, it is possible that the empirically constructed rules my contain non-clinically relevant logic. In order to avoid this situation we have established a process for auditing the best of the automatically created rule sets. The manually constructed rules are used as a basis for auditing. The generated rules are scanned for major deviations from the manual rule set and flagged for inspection. To date, the rules generated appear clinically viable. In small experiments with the worst case rule set less then 5% were suspicious. This highlights again that one of the outstanding benefits of using the fuzzy inference approach is the ability to "open the box" and clinically understand how the system performs.

The outstanding question that remains is exactly how this process can produce reasonable results in the presence of ambiguous data. This is especially puzzling in light of our earlier experiments using neural networks, and the difficult they had in converging to solutions. In part the fact that the genetic optimization process is essence a parallel stochastic search makes it very robust in terms of avoiding local minimums. In addition, the clinician annotated data bounds the problem sufficiently that truly unlikely combinations are removed from the gene pool relatively early. If the variation within the data is uncorrelated to the condition being tested, there would still remain many variations which would score high, but appear clinically unreasonable. Therefore we believe that we have successfully identified the correct inputs, and that they are capable of clearly indicating the condition we are concerned with, but that the shapes of the solution spaces are sufficiently convoluted that the genetic algorithm has an advantage over derivative based optimization techniques.

Acknowledgment

University of Michigan Medical Center Drs. O'Reilly and Rosenberg and Amy Shanks for clinical support and data collection.

References

[1] J.-S. R. Jang, C.-T. Sun, E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Upper Saddle Hall, NJ 07458, Prentice Hall, ch 7.

[2] Ibid, ch 12.

[3] T. Monk, V. Saini, B. Weldon, J. Sigl, "Anesthetic Management and One Year Mortality after Noncardiac Surgery", Anesthesia and Analgesia, vol. 100, 2005, pp. 4-10.

[4] A. Nurnberger, D. Nauck, R. Kruse, "Neuro-Fuzzy Control Based on the NEFCON-Model Under MATLAB/SIMULINK", online, http://fuzzy.cs.uni-magdeburg.de/~nuernb/wsc2/.

[5] J. Pendergast, et. al., "Anesthetic Overdose Input Parameter Prioritization", unpublished.

[6] D. Reich, et. al., "Predictors of hypotension after induction of general anesthesia", Anesthesia and Analgesia, vol. 101(3), 2005, pp. 622-8.

[7] Q. Xue, B. Taha, S. Reddy, T. Aufderheide, "An Adaptive Fuzzy Model for ECG Interpretation", presented at Marquette ECG Conference, 1998.