Fuzzy logic inference ruleset augmentation with sample data in medical decision-making systems

Alexander Kurochkin, Vasili Sadov
Belarussian State University
Minsk, Belarus
lawliet29@gmail.com, sadov@bsu.by

***Abstract*—Fuzzy inference systems are widely used in order to implement complex rule-based decision-making process in expert systems. One of their significant limitations, however, is the fact that rules themselves describe ambient semantic of decision-making process without taking real world data into account. This paper discusses the possible ways to use sample data in order to optimize fuzzy inference-based decision making.**

***Keywords*—expert systems, fuzzy logic, fuzzy inference systems, medical expert systems, machine learning**

I. Introduction

Expert systems are a useful tool for formalizing complex semantic decision-making processes performed by a domain expert in order to build a software platform that supports similar manner of decision-making. In a way, creating an expert system is a means to break down expert knowledge and experience into a set of formal semantic logical statements that allow inductive, deductive and abductive reasoning to be applied to real-world data. This, in turn, allows these rules to reach some kinds of conclusions based on semantic representation of input data, therefore generating new knowledge or producing new facts based on prior knowledge or existing facts. In a way, expert systems can be viewed as higher order formalizations of real-world data – not only do these systems require strict semantic formalizaiton of data, they also require a strict semantic formalization of any decision-making process associated with the data [1].

Inference-based expert systems use an inference engine as part of the decision-making process. These engines are expected to produce new facts based on intrinsic expert-specified semantic rulesets and some input facts. Essentially, inference process defines how input facts and rulesets are used to generate new facts and output data. In order to implement an expert system given a formal inference engine it is only required to form a corresponding semantic ruleset, containing a direct logical implication path from facts that are given to the expert system as an input to facts that can be used to determine problem output [2], [3].

One of the significant disadvanatages of most inference-based expert systems is the fact that only semantic ruels themselves serve as the ground truth for every decision made. While this is useful when dealing with some ambient decision-making process that cannot be formally verified with real-world data (for instance, because the data is scarce), this is not the case most of the applications. Usually, at least some measure of existing data can be obtained that binds input variables and the projected result. These data points, however, are mostly only used to check the correctness the decision-making process, and are not used to directly improve it.

The way rule-based expert systems operate is directly opposite to supervised machine learning approach. In supervised machine learning, decision-making process itself is not defined in any way. Instead, learning algorithm is expected, given a large set of data points with known outputs, to generate a decision-making model that infers data semantics and mimics the intrinsic input-output relationship. This approach yields great results when the direct semantic relationship between input and output parameters exists, but is not obvious; this semantic relationship can be expected to be determined during the learning process itself. However, the exact reasoning that led a fully-trained supervised machine learning model to produce specific output based on specific input is usually hard or impossible to determine, i.e. the decision-making process itself remains a black box; moreover, this decision-making process is expected to be different for different learning models and input data, the difference sometimes being very significant [4].

The aim of this paper is to discuss the possible ways to combine these two approaches – to employ supervised machine learning techniques in order to optimize and augment the expert system ruleset and its variable parameters to better fit existing sample data. The process is applied to fuzzy inference-based decision-making medical expert system for determining chorionicity.

II. Expert systems with fuzzy inference in medical applications

A key component of any expert system is inference engine. It defines, precisely, in what form expert knowledge should be presented to the system in order to support further decision-making process, i.e. it defines a formalization of the expert knowledge. Most commonly, inference engines require decision-making process to be described as a number of rules that are assumed to be truthful logical statements and can be utilized to generate new facts about existing data.

In general, the ruels used in inference engine are defined as a number of if-then rules, i.e. “IF (A) THEN (B)”. Logical statement A (antecedent) is usually some known fact, and logical statement B (consequent) is a fact that can be naturally deduced from A. Inference engines work by applying existing knowledge A to produce new knowledge B, i.e. by asserting the truth of consequents based on antecedent across the rulesets. The fact that consequent is true can be used as an output fact, i.e. as something that must be determined in context of the problem solved by expert system, or as prior knowledge for another rule, i.e. as part of antecedent statement for other rules, which, in turn, allows to generate more facts based on these rules with modus ponens inference. The process of cascading application of existing facts is known as chaining.

Most of the known inference engines employ either forward chaining or backward chaining in order to deduce the result. The primary difference between the two is the form of input to the system. Forward chaining means that problem input can be stated as a number of facts that are then used as antecedents in the ruleset, and the goal of the system is to deduce some of the consequent facts. Backward chaining means that problem input is not only a number of facts used in antecedents, but also a formal logical statement, and the goal of the system is to determine, based on given input facts, whether this statement is true according to existing ruleset. In general, backward chaining is harder to implement, but it allows expert systems to work as knowledge bases and use these input formal statements as a queries, implicitly performing a semantic analysis with existing facts. Such systems require non-linear traversal mechanism for rulesets in order to deduce which rules exatcly must be applied in order to verify the given statement. Forward chaining is usually easier to reason about, since it is only applied to deduce the correct facts based on known data provided as an input [2], [5].

One of the common approaches to building inference engines in expert systems is usage of fuzzy logic-based inference. Based on fuzzy algebra, fuzzy inference allows input facts to be stated not as a strict logical statements, but as fuzzy variables, i.e. with a specific degree of certainty. It enables a one-way formalization of concrete input parameter values to statements as fuzzy sets in order to form a fuzzy variable, that can be further used in logical statements of a ruleset to generate new data.

In a fuzzy inference system (FIS) any kind of numeric parameter can be used as input or output variable. From a higher order of applicaiton, FIS is a function - providing a projection of input parameters into output parameters. Input parameters are variables that are known prior, and output parameters are the unknown variables that must be determined by applying a set of rules.

When specifying rules in FIS, the elementary logical expressions used in them are generally defined not as a strict boolean value that can be true or false, but rather as a fuzzy variable that can have an arbitary degree of “truth-iness” from 0 to 1. In order to simplify the rule generation, all input variables can have several fuzzy sets associated with them that can be used as rules [3].

For instance, a medical expert system for determining chorionicity and amnionicity in multiple pregnancies can have a set of rules stated by the expert [6], [7]. One of them might look like this:

$$\begin{gathered}\text{IF } [(\textbf{amniotic membrane thickness}) \text{ IS ``Thick''}] \\ \text{AND } [(\textbf{duration}) \text{ IS ``End of the 1-st trimester''}] \\ \text{THEN } [(\textbf{chorionicity}) \text{ IS ``Likely dichorional''}]\end{gathered} \tag{1}$$

The rule (1) has 2 input variables (amniotic membrane thickness and duration) in antecedent and 1 output variable (chorionicity) in consequent. The antecedent is a compound statement - it has two simpler logical terms grouped with conjunction (logical “AND”). The first operand of conjunction is a statement [**(amniotic membrane thickness)** IS “Thick”]. In fuzzy inference systems all input and output parameters are assumed to have a crisp numeric value. For instance, one of the possible values for amniotic membrane thickness is 2.1mm, as measured during ultrasound inspection. But it is usually hard to operate with crisp values in the rules, because the terms in them allow for a degree of uncertainty. Here, we use the term “Thick”. By itself, the statement [**(amniotic membrane thickness)** IS “Thick”] actually defines a fuzzy set over crisp tickness values, in milimeters, which means that, for any crisp thickness value, it is possible to determine the “truth-iness” measure of this statement. For instance, for value of 2.1mm that statement might yield a 0.95 certainty fuzzy value. The exact mapping from crisp values to fuzzy certainity for each of the terms used in the rule is determined by a fuzzy membership function. Likewise, duration variable with the term “End of the 1-st trimester” is also a fuzzy set over the duration of the pregnancy, in weeks [6], [7].

The membership functions are usually selected from a range of basic functions, like triangular, trapezoid, sigmoidal, etc. The parameters of these functions can regulate slope, translation points and the general shape of the function. Typically, the parameters for membership functions are also defined by the expert. For instance, the fact amniotic membrane thickness is considered “Thick” at >2mm is part of the expert knowledge that is formalized within the system. The "gray" areas, however, where membership function takes values between 0 and 1, are usually interpolated between a known set of points linearly. For example, trapezoid membership function would be a common choice for "Thick" term, because there is a certainty that thickness over 2mm can be considered thick and below 1.7mm cannot be considered thick, so between 1.7mm and 2mm the values are linearly or sigmoidally interpolated.

Fuzzy inference process aggregates all the logical expressions in atecedents using a specific implementation of t-norm and t-conorm for AND and OR operators. Based on the output membership, all consequents that contain a specific output variable can be assigned a specific value. In classic Mamdani inference type system, resulting output values of the system are fuzzy sets. In order to produce a crisp value based on the fuzzy set, a centroid is most commonly used [2].

III. Fuzzy logic system parameters as machine learning optimization variables

As noted earlier, expert systems in general and fuzzy logic inference systems in particular only rely on their respective rulesets with a formal decision-making process in order to produce output values, and a significant disadvantage of such an approach is the fact that real datasets of the problem cannot be used to improve decision-making process. On the other hand, re-creating a decision-making process from scratch based on real data is the problem that machine learning aims to solve [1], [4].

In general, machine learning works based on the assumption that output data (the result of the decision-making process, in our case) can be modelled as a paremetric function:

$$\vec{y} = M(\vec{x}, \vec{\theta}), \tag{2}$$

where $\vec{y}$ is the output parameter vector, $\vec{x}$ is the input parameter vector and $\vec{\theta}$ is the model parameters vector. Given a number of data points with known input and output (a training set) $\{(\vec{x_i}, \vec{y_i})\}$, it is possible to calculate how well the model (2) fits these data points. The most commonly used metric is sum of squared differences (SSD) defined as follows:

$$S(\vec{\theta}) = \sum_{i=1}^{n} (M(\vec{x_i}, \vec{\theta}) - \vec{y_i})^2 \tag{3}$$

Other metrics that can be used include sum of absolute differences and mean values of squared error and absolute error, and other estimators based on median values. The general idea is that this metric can be described as "fitness", i.e. it numerically represents the ability of a particular model M with a concrete set of parameters $\vec{\theta}$ to generalize these data points [1].

The learning process itself is essentially an optimizing of any model performance metric, like (3), in regards to model parameters $\vec{\theta}$. The general idea is that variable parameters of the model $\vec{\theta}$ can be adjusted to make the model M behave in every possible way, and of those possible functions those are best used for a particular problem that guarantee that performance metric is at its minimum across all possible values of model parameters.

More complex models tend to fit the data better. However, the complexity of the model itself means that it performs poorly in generalization – additional non-linearity introduced into it works well for exact data points used for training, but the function itself can behave unpredictably in between these points. This problem is known as overfitting. If the model is overfit, its performance on the training set will be very good, but it will perform poorly on any known data items not used during training, and requires additional optimizations like parameter regularization [1].

As described earlier, machine learning, while able to fit the data for a variety of tasks, actually remains a black box even when properly trained. The decision-making process within, for instance, neural networks is based entirely on individual weight values. Sometimes it is possible to analyze the paths from input to output and determine how input features affect the output; however, most of the times this information doesn't shed light on how exactly the decisions are made. This is the main difference between classic expert systems and machine learning systems – the former work based on a formalized decision-making process without taking the real data into account, while the latter try to determine a set of parameters for some complex function so that it fits the real data without trying to formalize the decision-making proecss [4], [8].

Combinining fuzzy logic and neural networks to obtain the benifits of both have been explored in the past with the introduction of ANFIS (Adaptive Neuro-Fuzzy Inference System). However, the complexity of such systems mean that they remain a universal estimator and that their decision-making process is still obscure, since it depends heavily on training set [8]–[10].

In medical systems, real data for training is usually available in smaller quantities. At the same time, doctors that observe and diagnose the patients usually follow a set of generalist rules that aid with decision-making. These rules can be formed based on personal experience, or observing historical data, or taken from a well-known research on the topic, but the preference is usually given to methodology rather than statistics. For this reason, expert systems are a more natural choice for medical applications. However, historical data can and should be used not only to verify built systems, but also to help improve them. The radical improvement would mean the complete reorganization of the rulesets if, for instance, some of the rules can be disproven by a certain case. However, such decisions must be carefully weighted by the expert himself, since outliers in medical practice are a common occurance [6], [7].

The above means that incorporating real-world data statistical distribution as a basis for decision-making process, as is done in machine learning, is generally not a desireable approach. As such, a better way to use real data is to make smaller adjustments to the formal rule-based expert systems [10].

The formal parameters that can be optimized are rule weights and membership function values. It is important to note that both weights and membership function parameters must be constrained, because those parameters are part of the expert knowledge formalization, as well.

IV. Optimizing medical FIS for determining chorionicity with historical data

Given a ruleset of k rules, rule weight vector $\vec{w}$ of length k determines the scaling factor of this particular rule. The output membership of antecedent of i-th during antecedent aggregation are additionally multiplied by w_i.

Given a trapezoid membership function m_{trap} with parameters $a_1 < a_2 < a_3 < a_4$ defined as follows:

$$m_{trap}(x) = \begin{cases} 0, & \text{if } x \in (-\infty; a_1] \cup [a_4; \infty) \\ \frac{x-a_1}{a_2-a_1}, & \text{if } x \in (a_1, a_2) \\ 1, & \text{if } x \in [a_2; a_3] \\ \frac{a_4-x}{a_4-a_3}, & \text{if } x \in (a_3, a_4) \end{cases} \tag{4}$$

It is possible to treat a_1 - a_4 as model parameters per fuzzy membership, i.e. for every term of every input and output variable in the model. The outlines of membership functions (4) should still be defined by the expert, so it's usually helpful to formalize them as a set of predefined constraints $a_i^{(max)}$ and $a_i^{(min)}$ for these parameters and allow data optimization to variate parameters within these constraints.

A parameter vector for optimization $\vec{\theta}$ consists, as such, of rule weights w_i for every rule and trapezoid membership function parameters a_{ij} that lie within their respective constraints $a_{ij}^{(max)}$ and $a_{ij}^{(min)}$ for every term membership.

The only optimization strategy for such an algorithm is an iterative mesh descent, since function gradients cannot be approximated. This algorithm provides a very low guarantee of finding global minimum, but with existing decision-making model local minima already provide better results, as shown in the table I. The training set included 300 cases with known chorionicity with 10% used for cross-validation. Input parameters include results of various laboratory and examination reviews, based on which an expert was asked to provide a resolution. The results indicate that pure machine learning approach requires further tuning or model selection as it retains larger error in outlier cases, while fuzzy logic with optimization generally performs better than pure fuzzy inference, as predicted, approaching the expert estimation errors.

Table I
Training and validation errors for all predictions, by different approach

Approach	Training error	Validation error
Expert estimation	8.4%	N/A
Feedforward ANN	11.4%	19.6%
FIS	13.3%	N/A
FIS with optimization	9.3%	12.8%

V. Conclusion

Optimizing expert systems based on real-world data is a powerful way not only to verify the formal decision-making model, but to also augment it with statistical observations. This allows to retain the clarity of formal decision process, as formulated by an expert, while allowing the outliers present in live data to also be reflected in the model in a form of weights and membership function parameters. The results indicate that this approach yields a noticeable accuracy increase for fuzzy inference systems. Further studies, however, are required in order to optimize the augmentation process, since the only way of determining correct variable parameters is a full traversal, making it a costly and time-consuming process.

References

[1] I. H. Witten, E. Frank, M. A. Hall et al *Data Mining: Practical Machine Learning Tools and Techniques, 4th ed.*. Burlington, Morgan Kaufmann, 2016. 654 p.

[2] X. Wang, D. Nauck, M. Spott, R. Kruse. Intelligent data analysis with fuzzy decision trees. *Soft Computing*, 2007, vol. 11, no. 5, pp. 439-457.

[3] L. A. Zadeh Fuzzy sets. *Information and Control*, 1965, vol. 8, no. 3, pp. 338-353.

[4] R. A. Ghani, S. Abdullah, R. Yaakob Comparisons between artificial neural networks and fuzzy logic models in forecasting general examinations results. *International Conference on Computer, Communications, and Control Technology (I4CT)*, 2015, pp. 253-257.

[5] J. R. Quinlan Induction of Decision Trees. *Machine Learning*, 1986, vol. 1 pp. 81-106.

[6] O. Pribushenya, A. Kurochkin Otsenka platsentatsii pri mnogoplodnoi beremennosti s ispol'zovaniem sovremennykh ekspertnykh komp'yuternykh programm [Placentation evaluation of multiple pregnancies using modern expert computer programs]. *Sovremennye perinatal'nye meditsinskie tekhnologii v reshenii problem demograficheskoi bezopasnosti* [*Modern prenatal medical technologies in solving demographic safety problems*], 2017, vol. 10, pp. 106-111.

[7] A. Kurochkin, O. Pribushenya, V. Sadov Ekspertnaya meditsinskaya sistema po opredeleniyu khorial'nosti na osnove sistemy nechetkoi logiki [Expert medical system for chorionicity evaluation based on fuzzy logic system]. *Informatsionnye tekhnologii i sistemy* [*Information technologies and systems*], 2017, pp. 92-93.

[8] S. Rajab Handling interpretability issues in ANFIS using rule base simplification and constrained learning. *Fuzzy Sets and Systems*, 2018, in press.

[9] R. Mazouni, A. Rahmoun AGGE: A Novel Method to Automatically Generate Rule Induction Classifiers Using Grammatical Evolution. *Studies in Computational Intelligence*, 2015, vol. 570, pp. 270-288.

[10] E. Hüllermeier Does machine learning need fuzzy logic? *Fuzzy Sets and Systems*, 2015, vol. 281, pp. 292-299.

УТОЧНЕНИЕ НАБОРА ПРАВИЛ СИСТЕМЫ НЕЧЕТКОГО ВЫВОДА С ИСПОЛЬЗОВАНИЕМ ИСТОРИЧЕСКИХ ДАННЫХ В МЕДИЦИНСКИХ СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Курочкин А.В., Садов В.С.

Нечеткий вывод – широко распространённый подход к построению сложных процессов принятия решений в экспертных системах на базе набора правил. Одним из существенных недостатков таких систем является самостоятельное описание семантики процесса вывода без учета реальных данных. В работе рассматриваются возможные способы использования существующих исторических данных с целью оптимизации процесса принятия решений с использованием нечеткого вывода.

Received 09.01.19