Fuzzy Operator Trees for Modeling Rating Functions

Yu Yi, Thomas Fober, and Eyke Hüllermeier

Philipps-Universität

Marburg, Germany

Abstract

We introduce a new method for modeling rating (utility) functions which employs techniques from fuzzy set theory. The main idea is to build a hierarchical model, called a *fuzzy operator tree* (FOT), by recursively decomposing a rating criterion into sub-criteria, and to combine the evaluations of these sub-criteria by means of suitable aggregation operators. Apart from the model conception itself, we propose an evolutionary method for *model calibration* that fits the parameters of an FOT to exemplary ratings. The possibility to adapt an FOT to a given set of data makes the approach also interesting from a machine learning point of view.

1   Introduction

The evaluation and assessment of alternatives is a fundamental task and a key element in numerous types of applications. In this regard, the abstract concept of *utility*, mathematically formalized in terms of a *utility function*, has a longstanding tradition in economics where it plays an important role in the study of economic behavior [21, 13]. In fact, utility elicitation and the evaluation of alternatives are fundamental to decision making. Today, the advent of Internet technology has created a great interest in related topics as well, e.g., in the context of electronic markets, information retrieval systems, or search engines, just to mention a few.

In this paper, we propose a method for modeling utility functions, subsequently referred to as *Fuzzy Operator Trees* (FOT), that makes use of techniques from fuzzy set theory. A fuzzy operator tree implements a *fuzzy rating function*, that is, a utility function that maps to the range $[0, 1]$, where 0 corresponds to the lowest and 1 to the highest evaluation. Even though the original motivation comes from quality control, FOTs are completely general and widely applicable.

As will be detailed in Section 2, our approach allows a human expert to specify a model in the form of an FOT in a quite convenient and intuitive way. To this end, he simply has to split evaluation criteria into sub-criteria in a recursive manner, and to determine in which way these sub-criteria ought to be combined. The result of this process is the *qualitative structure* of the model. A second step, then, it to parameterize the model. To support or even free the expert form this step, we propose a method for *calibrating* the model on the basis of exemplary ratings. This method is introduced in Section 3 and evaluated empirically in Section 4. Related work will be discussed in Section 5.

2 Fuzzy Operator Trees for Modeling Rating Functions

Our approach is based on three important conceptions: (1) Modulation of measurements in terms of fuzzy sets; (2) hierarchical structuring of a rating function through recursive partitioning of criteria into sub-criteria; (3) flexible aggregation of sub-criteria by means of parameterized fuzzy operators.

2.1 Modulation of Features

At the lowest level, basic features (e.g., measurement values in product control) x_i are modulated in terms of associated fuzzy sets $F_i : \mathbb{R} \rightarrow [0, 1]$. Consequently, the actual input of the rating function is not a feature itself, but a basic evaluation thereof. This evaluation, $F_i(x_i)$, is a number in the unit interval, expressing to what extent x_i meets a basic constraint, e.g., the requirement to have a "desired" value that could be expressed in terms of a trapezoidal fuzzy set. The above approach allows one to treat different types of features X_i with different domains $\mathbb{X}_i$ in a unified way.

In principle, it thus also becomes possible to express restrictions on the *relation* between

different features $x_{i_1} \ldots x_{i_k}$, $k \leq n$. This can be done by means of a fuzzy relation

$$R : \mathbb{X}_{i_1} \times \ldots \times \mathbb{X}_{i_k} \to [0,1].$$

For example, the (fuzzy) constraint that "x_i and x_j are almost equal" can easily be formalized in terms of a fuzzy equivalence relation on $\mathbb{X}_i \times \mathbb{X}_j$.

In the most general form, the lowest level of our approach is thus given by a mapping

$$\phi : \mathbb{X}_1 \times \ldots \times \mathbb{X}_n \to [0,1]^m, \quad \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \mapsto \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_m \end{pmatrix} \tag{1}$$

where each ϕ_j, $j \in \{1 \ldots m\}$, is the membership degree of a projection $(x_{i_1} \ldots x_{i_{a_j}})$ of $(x_1 \ldots x_n)$ in a corresponding fuzzy relation $R_j : \mathbb{X}_{i_1} \times \ldots \times \mathbb{X}_{i_{a_j}} \to [0,1]$. The ϕ_j can be considered as derived features or, say, basic criteria.

2.2   Hierarchical Structuring of a Rating Function

Simplifying the overall evaluation of an alternative by rating different sub-criteria first and aggregating these ratings afterward is an intuitively appealing and commonly used strategy [25]. An extreme example from the field of utility theory is a *linear utility function* that combines the utility of sub-components by means of a weighted sum. Here, we propose a model that deviates from such functions in at least two respects: First, in addition to arithmetic combination, we also consider logical aggregation operators. Second, by decomposing criteria into sub-criteria in a recursive way, we allow for hierarchical structures of larger depth.

An example is shown in Fig. 1, where the assessment of a candidate for a certain job is decomposed into the criteria `formal skills` and `language skills`. The former criterion is in turn decomposed into skills in mathematics (`math`) and computer science (`CS`), the latter into `French` and `Spanish`. The basic "measurements" in this example could be given, e.g., by the school grade in the corresponding subject or the number of points in a language test.
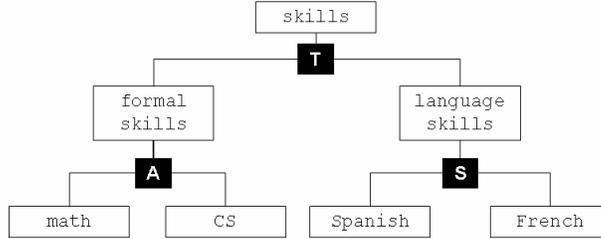
Figure 1: FOT for the assessment of a candidate.

2.3 Parameterized Fuzzy Operators

To aggregate the evaluation of sub-criteria, say, C_1 and C_2, into an evaluation of a criterion C, we make use of three types of operators that support different combination modes: t-norms and t-conorms (s-norms) [16] for conjunctive and disjunctive combination, respectively, and OWA (ordered weighted average) operators [26, 33] for an averaging combination. Recall that an OWA combination of k values $x_1 \dots x_k$ is defined by $A_w(x_1 \dots x_k) \stackrel{\mathrm{df}}{=} \sum_{i=1}^{k} w_i \cdot x_{\tau(i)}$, where τ is a permutation of $\{1 \dots k\}$ such that $x_{\tau(1)} \leq x_{\tau(2)} \leq \dots \leq x_{\tau(k)}$ and $w = (w_1 \dots w_k)$ is a weight vector such that $w_i \geq 0$, $i = 1 \dots k$, and $\sum_{i=1}^{k} w_i = 1$.

A conjunctive combination is the most stringent type of aggregation, while a disjunctive combination is the least stringent one. Within these two classes of operators, it is possible to make an even finer differentiation by looking at the order relation between t-norms (t-conorms). A t-norm T_1 (t-conorm S_1) is larger and, hence, less stringent than a t-norm T_2 (t-conorm S_2) if $T_1(a,b) \geq T_2(a,b)$ ($S_1(a,b) \geq S_2(a,b)$) for all $0 \leq a, b \leq 1$. The class of OWA operators "fills the gap" between the largest conjunctive combination, namely the minimum, and the smallest disjunctive combination, namely the maximum. We thus obtain a continuous spectrum of aggregation operators.

In the literature, several parameterized families of t-norms (t-conorms) have been proposed, such as the Dubois-Prade family [9]:

$$T_\lambda(a,b) = \frac{ab}{\max\{a,b,\lambda\}} \tag{2}$$

$$S_\lambda(a,b) = \frac{a + b - ab - \min\{a,b,1-\lambda\}}{\max\{1-a,1-b,\lambda\}} \tag{3}$$

The parameter λ can take values in the interval $[0,1]$. For $\lambda = 0$, (2) corresponds to the

4

minimum (and (3) to the maximum), while the product t-norm is obtained for $\lambda = 1$. We have adopted the family (2–3) for our approach, mainly because of its simplicity and the fact that a bounded parameter range (in this case $[0, 1]$) is quite convenient when it comes to calibrating the model.

Returning to our above example, the sub-criteria `formal skills` and `language skills` could be combined in a conjunctive way, since to qualify as a good candidate, both types of skills are considered important. To combine the ratings for mathematics and computer science, an averaging operator could be a reasonable choice. Finally, if it suffices to speak at least one foreign language, the skills in French and Spanish could be combined in a disjunctive way.

2.4 Weighing Sub-Criteria

In some cases, one may wish to weigh one sub-criterion higher than another one. To this end, we make use of *linguistic modifiers* [34, 17]. A linguistic modifier is a function mod : $[0, 1] \rightarrow [0, 1]$ that depicts the effect of linguistic hedges. For example, if mod : $x \mapsto x^2$ models the effect of VERY, and $a \in [0, 1]$ is the degree to which criterion C is satisfied, then $\text{mod}(a) \leq a$ is the degree to which VERY C is *satisfied.*

In the context of our application, a modifier mod : $x \mapsto x^\alpha$ with $\alpha > 1$ ($\alpha < 1$) can hence be used to increase (decrease) the importance of a sub-criterion in a conjunctive combination. The case of disjunctive combination is handled analogously.

2.5 Fuzzy Operator Trees

Putting everything together, an FOT is a tree in which every inner node corresponds to a fuzzy operator and every leaf node is given by a basic evaluation. More specifically, each inner node is marked by the type of operator, t-norm, s-norm, or averaging operator, (T, S, or A) along with its parameter(s). We mention that an FOT is not necessarily a binary tree; in fact, a node may have more than two successors, which is unproblematic as all three types of operators are associative. The edges of the tree are marked with the parameters (α) of the respective modifier functions. The root of the tree corresponds to the overall evaluation; see Fig. 2 for a schematic illustration.

The output of an FOT is a number in the unit interval reflecting the overall assessment of an alternative. In case a discrete category is preferred to a real-valued evaluation,
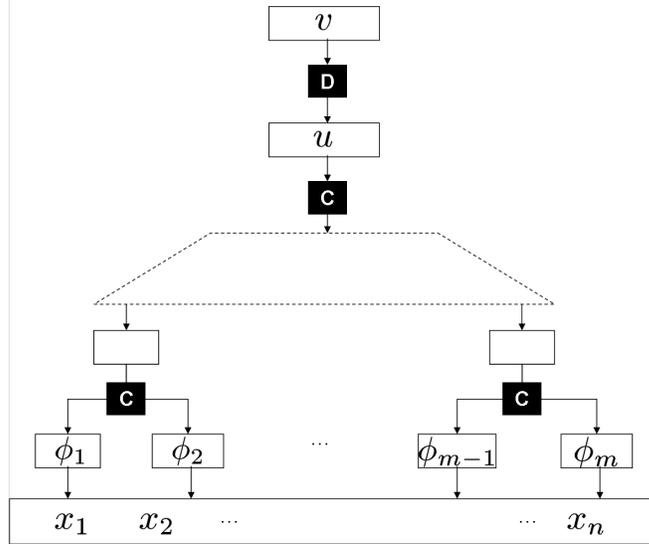
Figure 2: Schematic illustration of an FOT. The nodes marked with a "C" stand for one of the operators (T, S, or A). The continuous rating produced as an output is denoted u. The discretization operator, denoted by D, can be applied optionally to transform u into a discrete rating v.

a "discretizer" can be put on top of the tree, i.e., a function that transforms the real-valued output u into a discrete category v. For example, in quality control, the products are often put into categories such as A, B, and C. To realize such a discretization, we split the unit interval into ℓ sub-intervals defined by thresholds $t_j = j/\ell$, $j = 0 \ldots \ell$, and output the jth discrete output if $u \in \,]t_{j-1}, t_j]$.

2.6 Implementation

Even though implementation aspects are beyond the scope of this paper, we mention that we have implemented a tool that supports a user in modeling and using an FOT in a quite convenient way (see Fig. 3 for a screen shot). This tool offers a graphical user interface that allows one to add the components of an FOT by a simple mouse-click, to connect them, and to parameterize them. Besides, it offers a number of further convenient features, such as varying the level of detail (by hiding sub-trees or, the other way round, expanding the node of a criterion into its sub-criteria) or visualizing the strictness of aggregation operators (by using levels of green, yellow, and
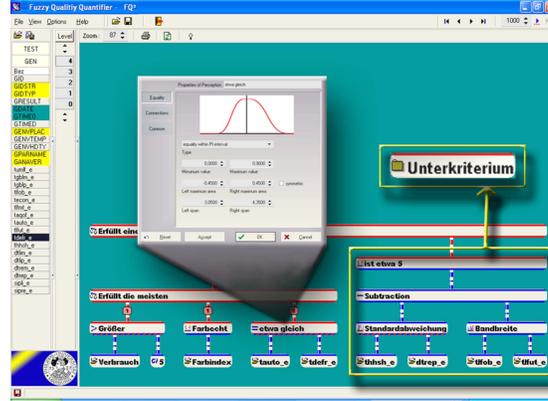
Figure 3: Screen shot of a tool for modeling FOTs.

red, respectively, for t-norms, OWA-operators, and t-conorms).

3 Calibration of Fuzzy Operator Trees

An FOT model consists of two parts: A *qualitative* part in the form of a node-labeled tree structure (where the label of a node identifies the type of aggregation operator), and a *quantitative* part that corresponds to the the parameters of the aggregation operators at the inner and the fuzzy sets at the leaf nodes of the tree. Often, a human expert will be able to provide the qualitative structure but may have problems with specifying precise parameters. In this section, we therefore address the problem to find a good parameterization in a *data-driven* way. To this end, we proceed from a set of training data in the form of exemplary ratings of alternatives. The goal, then, is to find parameters such that the model fits the data in an optimal way.

3.1 Problem Statement

Let all parameters of an FOT be collected in a vector θ:

- For every leaf node, θ includes the parameters specifying the associated fuzzy set (e.g., the center and support boundaries in the case of a triangular set);

- every inner node adds one or more parameters specifying the associated aggregation operator, namely $\lambda \in [0, 1]$ in the case of a t-norm (2) or t-conorm (3),

7

and weights $w_1 \ldots w_{k-1}$ in the case of an OWA operator with k inputs (w_k can be omitted due to the constraint $w_1 + \ldots + w_k = 1$);

– finally, there is one parameter α for every edge of the tree, specifying an associated linguistic modifier.

Denote by $\mathcal{M}[\theta]$ the FOT parameterized by θ, and by $\mathcal{M}[\theta](\mathbf{x})$ the output produced by this model for an input vector $\mathbf{x}$. Given a set of training data $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, the goal is to find

$$\theta^* \;=\; \arg\min_\theta \sum_{i=1}^{N} \ell\left(\mathcal{M}[\theta](\mathbf{x}_i) - y_i\right), \tag{4}$$

where $\ell(\cdot)$ is a suitable loss function. In the case of continuous outputs $y_i \in [0, 1]$, we use the squared error as a loss function, i.e., $\ell(\hat{y}_i, y_i) = (\hat{y}_i - y_i)^2$, which is a common choice for regression problems. The simplest loss function in the case of discrete outputs (also used in Section 4) is the 0/1-loss: $\ell(\hat{y}_i, y_i) = 0$ if $\hat{y}_i = y_i$ and $\ell(\hat{y}_i, y_i) = 1$.

3.2 A Calibration Method Using Evolution Strategies

The optimization problem (4) is extremely difficult, as it is highly *non-linear* and may involve *constraints* (e.g., relations between parameters specifying fuzzy sets). To solve it, we resorted to evolution strategies (ES), which are especially suitable for continuous optimization problems and produced the best results, not only within the class of evolutionary algorithms but also in comparison with other optimization methods that were tried, such as simulated annealing, ant colony optimization, etc. We refer [2] for a general introduction to ES.

Evolution strategies seek to optimize an objective function $f(\cdot)$ with respect to a set of configurable parameters; in our case, $f(\cdot)$ is given by the sum of prediction losses in (4), with θ as the parameter vector to be optimized. The fitness function, used to evaluate solution candidates, typically coincides with the objective function. In the case of categorical outputs, however, we slightly modified the fitness, since the simple 0/1-loss is problematic from an optimization point of view: It does not reward a modification of θ unless this modification changes the classification of at least one $\mathbf{x}_i$. That is, the fitness may remain unchanged even if θ is moved closer to the sought optimum θ^*, which leads to plateaus in the search space. To avoid this problem, we define the fitness function in terms of an alternative loss function suitable for ordinal classification. More
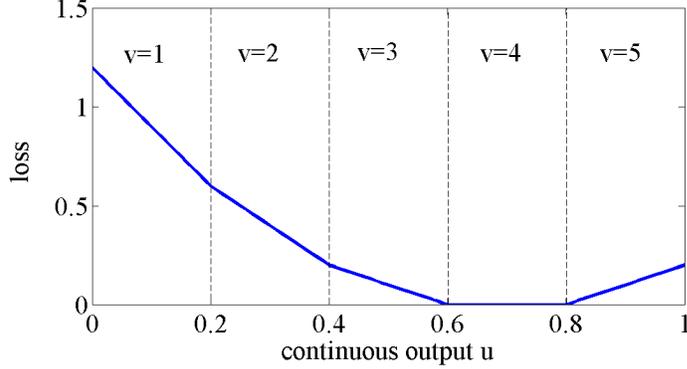
Figure 4: Example of the all threshold loss function (with $v = 4$ as the sought output).

specifically, the 0/1-loss in (4) is replaced by the "all-threshold" loss function $\ell(\cdot)$ which is defined as follows [23]: Let $0 = t_0 < t_1 < \ldots < t_k = 1$ denote the thresholds used for turning the continuous prediction $u \in [0,1]$ into a discrete output $v \in \{1, 2 \ldots k\}$. If the sought (discrete) output is $v = j$, then

$$\ell(u) = \begin{cases} 0 & \text{if} \quad t_{j-1} \le u \le t_j \\ \sum_{h=i}^{j-1} t_{h-1} - u & \text{if} \quad t_{i-1} \le u < t_i \le t_{j-1} \\ \sum_{h=i}^{j} u - t_h & \text{if} \quad t_j \le t_i < u \le t_{i+1} \end{cases} . \tag{5}$$

According to (5), $\ell(u) = 0$ is $u \in [t_{j-1}, t_j]$ and, therefore, the correct discrete output is predicted; otherwise, the loss is positive and increases with the distance of u from the interval $[t_{j-1}, t_j]$; see Fig. 4.

An ES is instantiated by several exogenous parameters that we optimized using the SPOT approach [1]; see Table 1 for the concrete parameters used in the experimental studies in Section 4. Since an ES is a stochastic algorithm, it produces results in a random way. Therefore, a *random restart* technique is often used, in which the whole algorithm is run k times, and the best among the results thus obtained is adopted. In our experiments in Section 4, we apply this technique with $k = 3$.

9

parameter	meaning	study 1	study 2
μ	population size	C	20
ν	selective pressure	2	2.5
ρ	recombination parameter	2	3
κ	maximal life-span of an individual	5	10
τ	learning parameter for the self-adaptation	$C^{-1/4}/\sqrt{2},$	$C^{-1/4}/\sqrt{2}$

Table 1: Exogenous parameters of an ES and concrete values used in the two experimental studies in Section 4; C is the complexity (number of nodes) of the trees to be calibrated in the first study.

4   Empirical Results

This section presents results of two experimental studies conducted to test our method for calibrating FOTs. The first experiment seeks to answer the question whether an ES is able to optimize FOTs of realistic complexity, and at the expense of acceptable runtime requirements. The goal of the second study was to compare FOTs with other model classes in a learning (classification) context in order to show that it can be usefully applied in cases where expert knowledge about the qualitative structure of an evaluation scheme is available.

4.1   First Experimental Study

As the possibility to conduct experiments in a controlled way was of utmost importance in this study, we made use of synthetic data. More specifically, we generated calibration problems of varying complexity in the following way: Given a fixed complexity in terms of a number C of nodes, an FOT is generated at random by starting with a root node and iteratively adding leaf nodes until the complete tree has C nodes. In each iteration, one among the current leaf nodes is randomly selected (using a uniform distribution) to become the predecessor of two additional nodes (we hence restrict to binary trees in this study). After the tree structure has been determined, the nodes are labeled with one of the three operators $(A,\ T,\ S)$, again at random with a probability of $1/3$ for each alternative, and each operator is parameterized with a random number r in the unit interval: in the case of T and S, r determines the parameter λ in (2–3), and in the case of an OWA operator, it instantiates the weight vector $w = (w_1, w_2)$ as $(r, 1 - r)$.
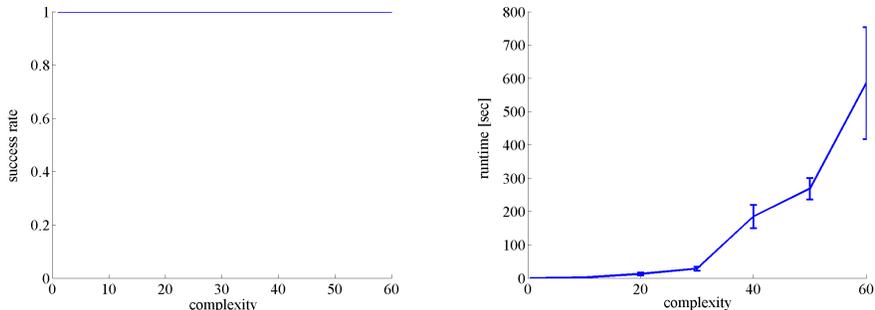
10

Figure 5: Results of the first experiment: mean and standard deviation of the success rate (left) and the runtime (right) as a function of the model complexity.

The parameter vector θ^* thus defined is the *ground-truth* that the EA is expected to rediscover.

Finally, $N = 200$ training examples[1] are produced by generating input vectors $\mathbf{x} \in [0,1]^m$ at random (again using a uniform distribution) and using the FOT to compute the corresponding (continuous) target outputs $y = \mathcal{M}[\theta^*](\mathbf{x})$. All experiments were carried out on a standard PC with Inter Duo-core 2.4 GHz CPU, 2G RAM, and Microsoft XP operating system.

The results, shown in Fig. 5, are quite promising. A calibration was considered a success if the cumulated error of the model $\mathcal{M}$ (defined by the best individual θ in the final population) was small enough or, more specifically, $\sum_{i=1}^{N} \ell(\mathcal{M}[\theta](\mathbf{x}_i) - y_i) < \varepsilon = 0.01$. As can be seen, the optimum solution is found rather reliably and without any exception, even for highly complex FOTs. The runtime does of course increase with the complexity and, as expected, the dependency seems to be super-linear. Nevertheless, the runtime is still acceptable in absolute terms, as a tree with 60 nodes (which is already a lot from a practical point of view) can be calibrated in about 10 minutes.

In a second experiment, we corrupted the sample outputs y_i by adding random noise (distributed normally with mean 0 and standard deviation σ). A calibration was now considered a success if $\sum_{i=1}^{N} \ell(\mathcal{M}[\theta](\mathbf{x}_i) - y_i) - \sum_{i=1}^{N} \ell(\mathcal{M}[\theta^*](\mathbf{x}_i) - y_i) < \varepsilon = 0.01$, where θ^* is the true parameter underlying the data generating process. As can be seen in Fig. 6, the results (shown for $\sigma = 0.01$ and $\sigma = 0.05$) are quite similar to the noise-free scenario, even though the success rate starts to deteriorate slightly for high

---

[1]These inputs can be thought of as the ϕ_i in (1); the fuzzy sets themselves are hence not needed.
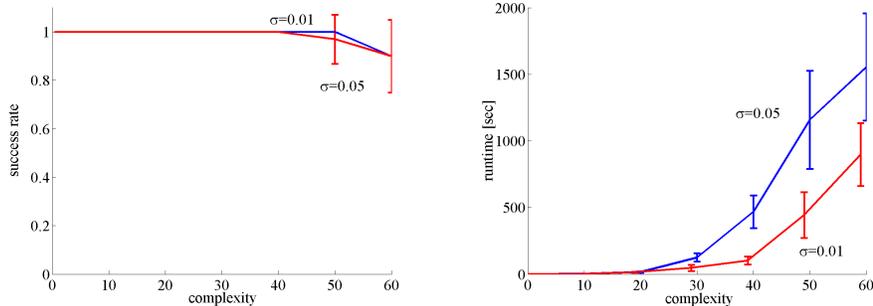
Figure 6: Results of the second experiment: mean and standard deviation of the success rate (left) and the runtime (right) as a function of the model complexity.
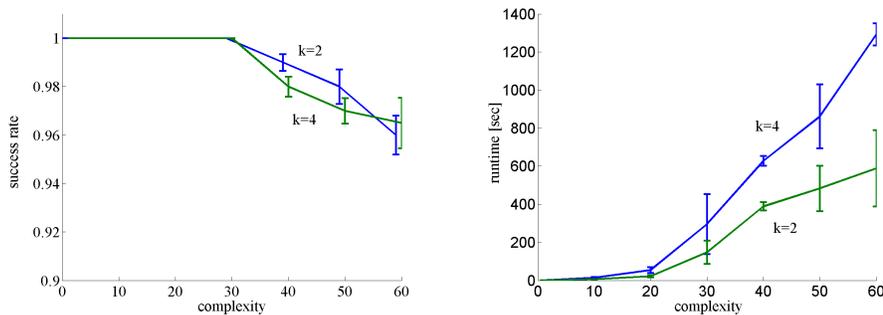

Figure 7: Results of the second experiment: mean and standard deviation of the success rate (left) and the runtime (right) as a function of the model complexity.

complexities.[2] Moreover, the results suggest that the level of noise (expressed in terms of the standard deviation σ) has a negative influence both on the success rate and the runtime.

In a third experiment, we investigated the case of categorical outputs. To this end, the continuous outputs were transformed into $k = 2$ and $k = 4$ discrete outputs, respectively, using an equi-width partition of the unit interval. Again, the success rate is very high and only starts to deteriorate slightly for high complexities, see Fig. 7 (note the rescaling of the ordinate in the left figure). Unsurprisingly, the calibration problem seems to become harder for larger numbers of categories, as can be seen from the increased runtime for $k = 4$.

---

[2]This deterioration necessarily occurs at some point, though it can be postponed by increasing the number of restarts.
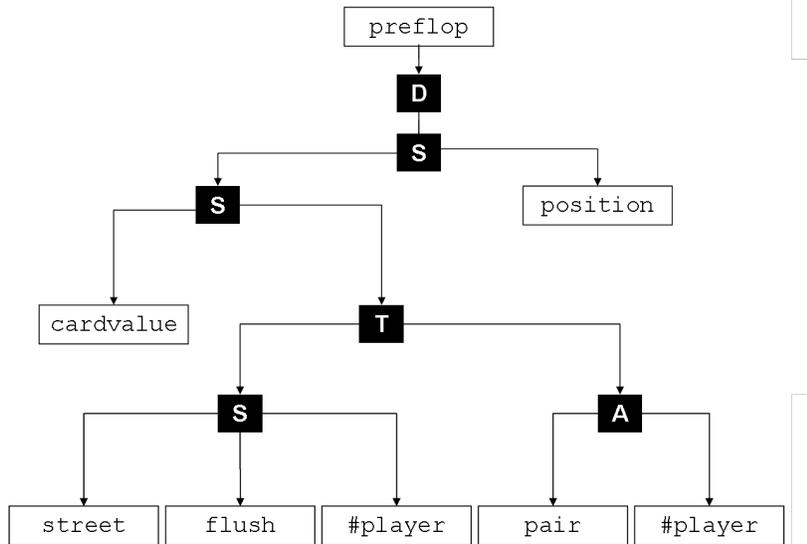
Figure 8: FOT for rating strategic situations in the game of poker.

In summary, the results of this experimental study show that our approach is able to cope with complex calibration problems, and that high-quality solutions can be achieved at the expense of an acceptable computational effort.

4.2 Second Experimental Study

In the second study, we considered an interesting type of rating problem, namely to evaluate strategic situations in the game of poker.[3] More specifically, we considered the pre-flop phase in the Texas Hold'em variant (see [27] for a general introduction), where a player can take one of two actions: `raise` or `call`. As a player will normally tend to raise in situations that appear to be favorable and call in less favorable ones, these two actions can be considered as discrete ratings, with `call` $\prec$ `raise`. From a machine learning point of view, the problem can obviously be seen as a binary classification task: Predict whether, in a given situation, a poker player will raise or call.

A situation is characterized in terms of the following numerical attributes/features:

- `cardvalue`: The quality of the hand (pair of cards) held by a player is specified

---

[3]Poker currently enjoys a great popularity in artificial intelligence research, especially in machine learning [12].

by the membership function $(c_1, c_2) \mapsto \gamma + (1 - \gamma)L(c_1, c_2)/8$, where (c_1, c_2) is a pair of cards, $L(c_1, c_2)$ its Sklansky group [28], and $\gamma \in [0, 1[$ is a parameter to be learned.

- `street`: The probability to extend the current hand to a street, using the (yet unknown) 3 cards that follow, specified by $(c_1, c_2) \mapsto p(c_1, c_2) \left(\max_{c,c'} p(c, c') \right)^{-1}$, where $p(c_1, c_2)$ is the probability to extend (c_1, c_2) to a street.

- `flush`, `pair`: The possibility to build a flush (pair), with a value of 1 if both cards have the same (different) color and γ otherwise, where $\gamma \in [0, 1[$ is a parameter to be learned.

- `#player`: Assuming a maximal number of 12 players, we defined the modulating fuzzy set by $n \mapsto 1 - \gamma(n - 2)/10$, where n is the actual number of players and $\gamma \in [0, 1]$ a parameter to be learned.

- `position`: The fuzzy set of "good positions" is defined by $p \mapsto \gamma + (1 - \gamma)p/\#\texttt{player}$, where p is the player's position and $\gamma \in [0, 1]$ a parameter to be learned.

Resorting to the above features and recalling our modest expertise in Poker, we designed the FOT shown in Figure 8. Needless to say, the features that we used will not completely determine the action of a poker player. Instead, a player may consider further aspects, different players have different strategies (e.g., bluffing plays an important role in poker), and eventually some randomness will also remain. As a result, data records consisting of the above attributes together with the player's action can be considered as extremely noisy, which makes the prediction problem difficult from a learning point of view.

Our data consists of 71,732 examples that we have extracted from the IRC Poker Database maintained by the University of Alberta Computer Poker Research Group.[4] Fixing the size of the training set by N, we randomly extracted N examples for training and used the rest of the data for testing. For every N, this was repeated 10 times. To compare the performance of our approach with state-of-the-art classifiers, we included C4.5 [22] and RIPPER [5].[5]

The results in terms of the average and standard deviation of the classification rate, as a function of the number N of training examples, are shown in Fig. 9. One of the most

---

[4] `http://games.cs.ualberta.ca/poker/IRC/`
[5] We used the implementations offered by the machine learning framework WEKA [32].
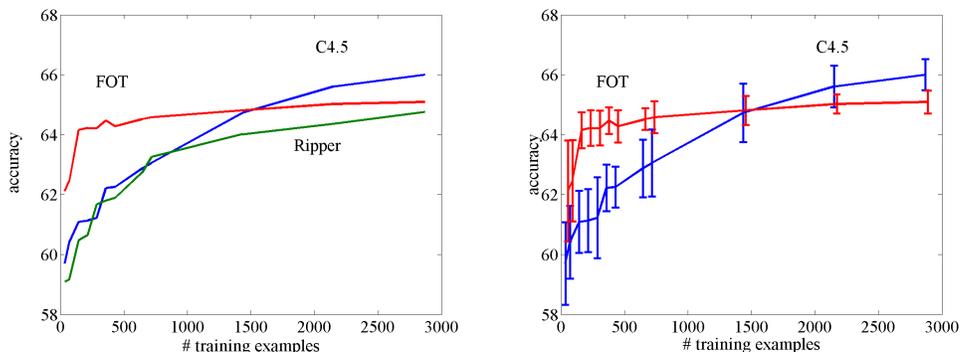
Figure 9: Left: Average accuracy of classifiers in the poker application; right: standard deviations of the classifiers (Ripper omitted for the sake to clearness).

interesting aspects of the results concerns the fact that, if relatively few training data is available, FOT significantly outperforms the other methods, while this superiority decreases and eventually disappears with an increasing size of the training data. In fact, the improvement of FOT is drastic for the first few training examples, whereas the rest of the examples does not contribute very much. Besides, the standard deviation of the accuracy is significantly smaller for FOT than for the other models (see right graphic in Fig. 9).

To explain this finding, note that our FOT has a very strong *inductive bias*, due to the fact that it has a fixed, predetermined structure. Compared to this, the other models are much more *flexible*. Now, it is well-known that, when learning models from data, the high flexibility of a model class comes along with a risk of *overfitting*, which is especially high for small data sets and becomes less severe for larger ones. Therefore, a strong inductive bias is especially useful if data is not abundant. However, at least if the bias is not perfect, it may also become hindering, namely when enough data is available to extract an optimal hypothesis even from a very rich model class; in this case, the bias may produce a problem of *underfitting*. Exactly this phenomenon seems to be responsible for the results that we obtained. These results again confirm that our approach is especially useful if reliable background knowledge is available, as it allows one to formalize and exploit such knowledge in a convenient way, while alternative approaches may otherwise be preferable.

Finally, let us again bring up the aspect of interpretability, which we consider as another strong advantage of our approach. In fact, even though the C4.5 models (just like the

Ripper classifiers) achieve a slightly higher classification rate if enough training data is available, these models are quite complex and difficult to understand: The size of the decision trees tends to increase with the size of the training data (in contrast to our FOT, which has a fixed number of 5 inner nodes) and can easily contain as many as 90 inner nodes; besides, the splitting conditions are often non-intuitive. While such models might still be acceptable to predict actions in Poker, interpretability becomes much more an issue in critical applications such as quality control or classification in medicine. In these fields, a human expert will probably not trust in a decision tree model with 90 inner nodes.

5   Related Work

The method presented in this paper is related, in one way or the other, to several research fields and existing approaches therein. In particular, such relations are established by (i) the idea of hierarchical modeling, (ii) the use of fuzzy logic-based operators in decision making, (iii) the calibration of a network-like structure to given input/output data.

The idea of hierarchical modeling is of course not new. In the field of utility modeling and decision making, the perhaps most well-known approach in this regard is the Analytical Hierarchy Process (AHP) [25]. AHP is a decision making technique that supports the rational evaluation of advantages and disadvantages concerning different alternative solutions to a *multi-objective* decision problem. Like our approach, it is based on a hierarchical structure, i.e., a recursive splitting of criteria into sub-criteria. However, the sub-criteria are combined in a different way. Moreover, the calibration of the structure is completely different, as it is based on the pairwise comparison between criteria. The comparisons have to be made by the user to figure out the relative importance (weight) of criteria. To check the consistency of the user responses, the matrix obtained by the pairwise comparisons is evaluated on the basis of its eigenvalues. Difficulties related to the pairwise comparison approach have been a focus of criticism of the AHP (see e.g. [6]).

Closer to our approach is the method proposed in [3], that induces a hierarchical structure by analyzing "functional dependencies". To aggregate criteria at interior nodes, the authors make use of decision rule tables that can be obtained in a data-driven way. Unfortunately, such tables can be difficult to understand, so the whole approach can be

criticized from an interpretability point of view. Besides, the decision tables that have to be stored at each interior node may become complex in the case of large datasets.

Regarding point (ii) above, a huge amount of work has been done in the field of fuzzy preference modeling and multi-criteria decision making (e.g. [8]), a review of which is clearly beyond the scope of this paper. To the best of our knowledge, however, an approach directly comparable to our FOTs, including a means for model calibration, has not been proposed so far.

The issue of model selection and parameter estimation has been addressed, though, for simpler types of decision models, especially for models using a single aggregation operator. For example, the problem of fitting parameters on the basis of exemplary outputs has been studied for weighted mean and OWA operators [7, 29], the WOWA (weighted OWA) operator [30], the Choquet integral [18, 31], and the Sugeno integral [14]. Besides, attempts have been made to identify the parameters of such models using other types of information, such as the so-called "orness" or degree of disjunction [10, 11] as well as preferences and order relations [4, 19].

With respect to point (iii), the calibration of an FOT may of course remind of related hierarchical or graph-based models such as artificial neural networks (ANN). There are, however, important differences between our approach and common ANN models such as multilayer perceptrons. In particular, ANNs typically have a single type of node (associated with an activation function), and only the weights on the edges are adapted. Moreover, the topology is usually not a tree; instead, in feed-forward nets, two successive levels are fully connected. Finally, ANNs do not offer an obvious logical interpretation. This latter disadvantage is to some extent avoided by neuro-fuzzy systems [20] and related approaches such as ANFIS (Adaptive Network-Based Fuzzy Inference System [15]), in which activation functions are replaced by logical operators. The other differences mentioned above, however, still remain also for these approaches.

6 Summary and Conclusion

This paper has introduced fuzzy operator trees as a convenient tool for modeling rating (utility) functions. The key idea of this approach, which appears to be appealing from a modeling point of view, is to express a rating function in terms of a hierarchical structure by decomposing criteria into sub-criteria in a recursive way. The evaluations

of sub-criteria can be combined by means of aggregation operators of different character: conjunctive, averaging, and disjunctive.

To support a human modeler in designing FOTs, we have developed a calibration method based on evolution strategies that fits the parameters of a (qualitative) model structure to a given set of training data. The empirical results that we obtained are rather promising and show that the method is powerful enough to be used in practice.

Several interesting issues remain to be addressed in future work. In particular, we plan to elaborate on the idea of using FOTs as a machine learning tool, especially to solve ordinal classification problems. In fact, there are many applications of learning methods, for example in the field of recommender systems [24], in which the model to be induced can be considered as a kind of (discrete) utility function. The ability to easily incorporate background knowledge, as it can be done by specifying an FOT, appears to be especially advantageous for applications of this type. Nevertheless, from a machine learning point of view, it may also be of interest to adapt the structure of an FOT, or at least parts thereof, in addition to the model parameters. One idea to approach this problem is to combine our evolution strategies with a discrete optimizer, such as genetic algorithms.

References

[1] Thomas Bartz-Beielstein. *Experimental Research in Evolutionary Computation - The New Experimentalism.* Natural Computing Series. Springer Verlag, Heidelberg, Berlin, 2006.

[2] H.G. Beyer and H.P. Schwefel. Evolution strategies – a comprehensive introduction. *Journal Natural Computing*, 1(1), 2002.

[3] M. Bohanec and V. Rajkovic. Knowledge acquisition and explanation for multi-attribute decision making. *In 8th Int. Workshop on Expert Systems and their Applications*, 1:59–78, 1988.

[4] E.U. Choo and W.C. Wedley. Optimal criterion weights in repetitive multicriteria decision making. *The Journal of the Operational Research Socienty*, 36(11):983–992, 1985.

[5] W.W. Cohen. Fast effective rule induction. In *Proc. 12th International Conference on Machine Learning*, pages 115–123, Tahoe City, CA, 1995. Morgan Kaufmann.

[6] J.S. Dyer. Remarks on the analytic hierarchy process. *Management Science*, 36(3):249–258, 1990.

[7] D.P. Filev and R.R. Yager. On the issue of obtaining OWA operator weights. *Fuzzy Sets and Systems*, 94:157–169, 1998.

[8] J. Fodor and M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[9] J. Fodor and R.R. Yager. Fuzzy set-theoretic operators and quantifiers. In D. Dubois and H. Prade, editors, *Fundamentals of Fuzzy Sets*, pages 125–194. Kluwer Academic Publishers, Boston/London/Dordrecht, 2002.

[10] R. Fuller and P. Majlender. An analytic approach for obtaining maximal entropy OWA operator weights. *Fuzzy Sets and Systems*, 124:53–57, 2001.

[11] R. Fuller and P. Majlender. On obtaining minimal variability OWA operator weights. *Fuzzy Sets and Systems*, 136:203–215, 2003.

[12] J. Fürnkranz and M. Kubat, editors. *Machines That Learn to Play Games*. Number 8 in Advances in Computation: Theory and Practice. Nova Biomedical, 2002.

[13] P. Gärdenfors and N.E. Sahlin, editors. *Decision, Probability, and Utility*. Cambridge University Press, 1988.

[14] K. Ishii and M. Sugeno. A model of human evaluation process based on fuzzy measure. *International Journal of Man-Machine Studies*, 22:19–38, 1985.

[15] J.S.R. Jang. ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 23:665–685, 1993.

[16] EP. Klement, R. Mesiar, and E. Pap. *Triangular Norms*. Kluwer Academic Publishers, 2002.

[17] G. Lakoff. Hedges: a study in meaning criteria and the logic of fuzzy concepts. *Journal of Philosophical Logic*, 2:458–508, 1973.

[18] J.L. Marichal and M. Roubens. Determination of weights of interacting criteria from a reference set. *European Journal of Operational Research*, 124:641–650, 2000.

[19] P. Meyer and M. Roubens. Choice, ranking and sorting in fuzzy multiple criteria decision aid. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 471–506. Springer-Verlag, 2005.

[20] D. Nauck, F. Klawonn, and R. Kruse. *Foundations of Neuro-Fuzzy Systems*. Chichester, UK, 1997.

[21] J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. John Wiley and Sons, 1953.

[22] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[23] J.D.M. Rennie and N. Srebro. Loss functions for preference levels: Regression with discrete ordered labels. In *Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling*, Edinburgh, Scotland, 2005.

[24] P. Resnik and H.R. Varian. Recommender systems. *Communications of the ACM*, 40(3), 1997.

[25] T.L. Saaty. *The Analytic Hierarchy Process*. McGraw-Hill, 1980.

[26] B. Schweizer and A. Sklar. *Probabilistic Metric Spaces*. North-Holland, New York, 1983.

[27] D. Sklansky. *Hold'em Poker: A Complete Guide to Playing the Game*. Two Plus Two, 1996.

[28] D. Sklansky and M. Malmuth. *Hold 'em Poker for Advanced Players*. Non Basic Stock Line, 3 edition, 1999.

[29] V. Torra. On the learning of weights in some aggregation operators. *Mathware and Soft Computing*, 6:249–265, 1999.

[30] V. Torra. OWA operators in data modeling and reidentification. *IEEE Transactions on Fuzzy Systems*, 12(5):652–660, 2004.

[31] Z. Wang, K.S. Leung, M.L. Wong, J. Fang, and K. Xu. Nonlinear nonnegative multiregressions based on Choquet integrals. *International Journal of Approximate Reasoning*, 25:71–87, 2000.

[32] IH. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, San Francisco, 2 edition, 2005.

[33] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE *Transactions on Systems, Man, and Cybernetics*, 18(1):183–190, 1988.

[34] L.A. Zadeh. A fuzzy-set theoretic interpretation of linguistic hedges. *J. Cybernetics*, 2(3):4–32, 1972.