On the Application of Preaggregation Functions to Fuzzy Pattern Tree

Pan Su[a b], Tianhua Chen[c], Haoyu Mao[a], Jianyang Xie[a], Yitian Zhao[a*] Jiang Liu[d a]

[a]*Cixi Institute of Biomedical Engineering, Chinese Academy of Sciences, China*
[b]*School of Control and Computer Engineering, North China Electric Power University, China*
[c]*Department of Computer Science, School of Computing and Engineering, University of Huddersfield, UK*
[d]*Department of Computer Science and Engineering, Southern University of Science and Technology, China*

Abstract—**Building transparent knowledge-based systems in the form of accurate and interpretable fuzzy rules is one of the significant applications of fuzzy set theory. The fuzzy connectives, i.e., T-norm/conorm, play the role of connecting fuzzy sets, which are essentially linguistic terms extracted from the knowledge embedded in a given data set. Fuzzy pattern tree is a recently proposed novel machine learning technique, which grows a hierarchical binary tree for each known class utilising conventional T-norms/conorms and aggregation operators. Preaggregation functions are recently proposed in the literature as a type of generalised aggregation functions, which have achieved successes in a number of applications. This paper proposes a preaggregation-based approach with application to the construction of fuzzy pattern tree. An experimental study is done to explore the performance of the fuzzy pattern tree where preaggregation functions are employed in comparison to that where conventional aggregation operators are utilised. Experimental results demonstrate that the performance of fuzzy pattern tree incorporated with the preaggregation function generated by Nilpotent minimum T-norm outperforms those with alternative preaggregation functions and the commonly used ordered weighted averaging operators.**

Index Terms—**preaggregation, aggregation, fuzzy pattern tree, knowledge base, triangular norm**

I. Introduction

Knowledge-based systems (KBSs) are aimed at representing knowledge explicitly via tools such as production or if-then rules, which allow a system to reason about how the conclusion is derived and to provide explanation of its reasoning to the user [1]. In case of situations where the information such as data or knowledge is imprecise in nature, fuzzy systems have been regarded as a popular and effective means to build KBSs [2]. Fuzzy KBSs are able to deal with vague concepts that are ubiquitous in natural languages and practical reasoning, which supports the design and implementation of such KBSs in effectively addressing real-world problems. Many approaches [3], [4], [5], [6] have been proposed for generating and learning fuzzy KBSs to represent the input/output behaviour of a certain problem, including the development of fuzzy rule-based classification systems such as fuzzy decision trees [7], [8].

Triangular norms (T-norms) generalising intersection in the framework of fuzzy metric spaces are essential components in a number of computational problems [9]. Most of the existing fuzzy rule induction methods focus on searching rules using the classic T-norms only. Different from the traditional rule induction methods that usually utilise only $\min$ and $\max$ operators, a fuzzy pattern tree is proposed in [10] as a novel machine learning technique and has been applied to the fields of classification, regression [11], and learning with data streams [12]. The structure of a fuzzy pattern tree is a binary tree whose inner nodes are marked with fuzzy logical and arithmetic operators. Apart from the $\min$ and $\max$ operators, it also utilise operators such as Łukasiewicz T-norms/conorms, Weighted Averaging (WA), Ordered Weighted Averaging (OWA), and so on for assembling knowledge pieces.

While being significant in the construction of pattern trees, a key property for defining aggregation operators lies in monotonicity and boundary conditions. Recently, a type of generalised aggregation functions by relaxing the monotonicity to directional monotonicity is proposed and termed preaggregation functions, which have already made success in several applications [13], [14]. Following on this promising trend, this paper presents an initial work to incorporate preaggregation functions to the construction of fuzzy patter trees. A set of preaggregation functions are generated by the Choquet integral and symmetric measures for knowledge extraction and inference in classification problems. The proposed construction of fuzzy pattern trees extends fuzzy pattern trees wherein monotonic operators have been applied to construct fuzzy propositions. The preaggregation-based models are compared with those on the basis of conventional aggregation functions on a number of UCI data sets. Experimental results demonstrate the suitability of using preaggregation functions to improve accuracy of fuzzy pattern trees in classification.

II. Preliminaries

A. Aggregation Functions

Triangular norm (T-norm) is a bivariate operator commonly used in the framework of fuzzy set and fuzzy logic. It can be deemed as generalisation of intersection and conjunction in fuzzy set and logic, respectively. A T-norm [15] is a mapping

*Corresponding author (e-mail: yitian.zhao@nimte.ac.cn) This paper is supported by the National Natural Science Foundation of China (61601029), Zhejiang Provincial Natural Science Foundation of China (LZ19F010001), Natural Science Foundation of Ningbo (2018A610055).

$T : [0,1] \times [0,1] \rightarrow [0,1]$ which for all $x, y, z, x', y' \in [0,1]$ satisfies:

1) commutativity: $T(x,y) = T(y,x)$;
2) monotonicity: $T(x,y) \leq T(x',y')$, if $x \leq x'$ and $y \leq y'$;
3) associativity: $T(x, T(y,z)) = T(T(x,y), z)$; and
4) boundary condition: $T(x,1) = x$.

A number of T-norms have been proposed in the literature such as:

- the minimum T-norm: $T_{\min}(x,y) = \min(x,y)$;
- the algebraic product T-norm: $T_{\text{p}}(x,y) = x \cdot y$;
- the Łukasiewicz's T-norm: $T_{\text{Ł}}(x,y) = \max(x+y-1, 0)$;
- the Nilpotent minimum T-norm: $T_{\text{Nil}}(x,y) = \max(x + y - 1, 0)$, if $x + y > 1$; $T_{\text{Nil}}(x,y) = 0$, otherwise; and
- the Hamacher Product T-norm: $T_{\text{Ham}}(x,y) = 0$, if $x = y = 0$; $T_{\text{Ham}}(x,y) = \frac{xy}{x+y-xy}$, otherwise.

It is worth noting that since T-norms satisfy associativity, they can be extended to n-ary operators. A function $A : [0,1]^n \rightarrow [0,1]$ is an n-ary aggregation function if it satisfies:

1) $A(0, \cdots, 0) = 0$, $A(1, \cdots, 1) = 1$, and
2) for each $i \in \{1, \cdots, n\}$, if $x_i \leq y$, then $A(x_1, \cdots, x_n) \leq A(x_1, \cdots, x_{i-1}, y, x_{i+1}, \cdots, x_n)$.

Ordered Weighted Averaging (OWA) operators form a family of aggregation procedures which may be seen as a special type of weighted average based on the ordering of their arguments. The fundamental property of this family of operators is the reordering step in which the arguments are rearranged in descending order and subsequently integrated into a single aggregated value.

A mapping $A_{owa} : \mathbb{R}^n \rightarrow \mathbb{R}$ is called an OWA [16] aggregation if

$$A_{owa}(x_1, \cdots, x_n) = \sum_{i=1}^{n} w_i x_{(i)}$$

where $x_{(i)}$ is a permutation of $x_i \in \mathbb{R}, i \in \{1, \cdots, n\}$, which satisfies that $x_{(i)}$ is the i-th largest amongst x_i, and $w_i \in [0,1]$ is a collection of weights that jointly satisfy $\sum_{i=1}^{n} w_i = 1$.

The most commonly seen aggregation functions include the arithmetic mean, $\max$, and $\min$, which can be treated as special cases of OWA. In particular, the arithmetic mean can be obtained by setting $w_i = 1/n$ for each $i \in \{1, \cdots, n\}$, the $\max$ by $w_1 = 1$ and $w_i = 0$ for $i \neq 1$, and the $\min$ by $w_n = 1$ and $w_i = 0$ for $i \neq n$. The aggregation behaviour of OWA operators is bounded by the max and min operator as: $\min\{x_1, \cdots, x_n\} \leq A_{owa}(x_1, \cdots, x_n) \leq \max\{x_i, \cdots, x_n\}$.

B. Preaggregation Functions

The problem of relaxing the monotonicity in the definition of aggregation has recently attracted a lot of interest. The notion of weak monotonicity is proposed in [17] where monotonicity is required only along the direction of first quadrant diagonal. This concept of weak monotonicity has been further extended in [18] by introducing the notion of directional monotonicity, which allows monotonicity along fixed directions. In particular, directionally monotone functions are generalisation of both weak monotone functions and aggregation functions.

A function $P : [0,1]^n \rightarrow [0,1]$ is an n-ary preaggregation function if P satisfies:

1) $P(0, \cdots, 0) = 0$, $P(1, \cdots, 1) = 1$, and
2) there exists a real vector $r = (r_1, \cdots, r_n) \in [0,1]^n$ such that for all $(x_1, \cdots, x_n) \in [0,1]^n$ and $(x_1 + \alpha r_1, \cdots, x_n + \alpha r_n) \in [0,1]^n$,$\alpha > 0$, it satisfies:

$$P(x_1 + \alpha r_1, \cdots, x_n + \alpha r_n) \geq F(x_1, \cdots, x_n).$$

If P is a preaggregation function with respect to a vector r, P is called r-preaggregation function. Three methods of constructing preaggregation functions are introduced in [19], including: the composition of appropriate functions, the construction of the discrete Choquet integral, and construction of the discrete Sugeno integral. In this paper, the Choquet integral is employed to generate preaggregation functions.

The Choquet integral combines the inputs in such a way that the importance of different groups of inputs (coalitions) may be taken into account. Let $N = \{1, 2, \cdots, n\}$ and n is an arbitrary positive integer n, a function $m : 2^N \rightarrow [0,1]$ is a fuzzy measure if for all $X, Y \subseteq N$, it satisfies: if $X \subseteq Y$, then $m(X) \leq m(Y)$; and $m(\emptyset) = 0$ and $m(N) = 1$. For example, the symmetric measure is defined as: given $W = (w_1, \cdots, w_n) \in [0,1]^n$ such that $\sum_{i=1}^{n} w_i = 1$, for any nonempty subset $A \subseteq N$, $m_s^W(A) = \sum_{i=1}^{|A|} w_i$. The discrete Choquet integral of $x = (x_1, \cdots, x_n) \in [0,1]^n$ with respect to m is defined as a function $C_m : [0,1]^n \rightarrow [0,1]$, given by

$$C_m(x) = \sum_{i=1}^{n} (x_{(i)} - x_{(i-1)}) \cdot m(A_{(i)}) \tag{1}$$

where $x_{(1)}, \cdots, x_{(n)}$ is an increasing permutation on the input x, that is, $0 \leq x_{(1)} \leq \cdots \leq x_{(n)}$, with the convention $x_{(0)} = 0$, and $A_{(i)} = \{(i), \cdots, (n)\}$ being the subset of indices of $n - i + 1$ largest components of x. The OWA aggregation is a special case of the Choquet integral, with respect to symmetric fuzzy measures m_s^W [20].

III. Application of Preaggregation Functions to Fuzzy Pattern Tree Construction

One of the most advantages of fuzzy systems lies in the interpretability as they are able to perform inference with human readable knowledge expressed in form of fuzzy if-then rules [21]. The explicit presentation of knowledge helps users to gain insights into the complex problems and to facilitate the explanation of their solutions. The induction of fuzzy decision tree is often considered as one of the most common techniques for the acquisition of interpretable fuzzy rule-based systems.

The construction of a fuzzy decision tree has been inspired following the fashions adopted in generating a crisp counterpart such as the ID3 [22]. Take the classic fuzzy ID3 as an example [7], which is constructed from the root node representing a fuzzy set of instances. The conjunctions between fuzzy terms along the current branches are calculated from the root to leaf nodes. The criteria of selecting a particular feature to split an inner node is based on the concept of fuzzy entropy. The tree stops growing till the ambiguity of each leaf is smaller than a threshold. Despite the recently proposed fuzzy pattern tree also utilises tree-based hierarchical structure, the meanings of the associated nodes and branches completely differ from those of conventional decision tress. Particularly, the root of a fuzzy pattern tree represents a fuzzy set of instances which is labelled as one class. Its leaves represent fuzzy terms and inner nodes represent fuzzy aggregations.

A. Top-down Construction of Fuzzy Pattern Trees

A fuzzy pattern tree [23] is a binary tree with each leaf L associated with a fuzzy proposition such as 'Age is *young*', where 'Age' is a linguistic variable that takes the value of a fuzzy linguistic term, i.e, '*young*' in this case. As 'Age' is considered a real-valued feature in practice, the so-called fuzzification process makes it possible to take linguistic terms such as '*young*' and '*old*' as values, which are essentially functions defined on its domain. The fuzzy proposition consisted of a linguistic variable and the corresponding fuzzy set is therefore one of the basic information granules as input to grow a fuzzy pattern tree.

On the other hand, each inner node of a fuzzy pattern tree is associated with a bivariate aggregation function $A : [0, 1]^2 \rightarrow [0, 1]$. A new fuzzy set I is formed at each inner node by applying A to its two children I_a and I_b as: $\mu_I(p) = A(\mu_{I_a}(p), \mu_{I_b}(p))$, where $p, p \in P$ is a single instance out of the data set P and $\mu_I(p)$ is the membership of p to I. Following the same manner, a complete fuzzy pattern tree predicts the degree of an instance belonging to its underlying class by calculating off the aggregations represented by inner nodes on the paths from the leaves to the root.

Formally, let R represent the root of a fuzzy pattern tree for a certain class C. Through matching against the underlying pattern tree, $\mu_R(p)$ indicates the membership of instance p belongs to class C predicted by that tree. The true membership of instance p to the class C is represented as $\mu_C(p)$. In crisp case where the instance label is deemed to overlap with one concept, $\mu_C(p)$ is binary such that $\mu_C(p) = 1$ indicates $p \in C$, otherwise $\mu_C(p) = 0$. For the targeting class C and an output fuzzy set at the root of a tree R, the error is empirically evaluated by the Rooted Mean Squared Error (RMSE) in this paper among alternative options [10] [23]:

$$RMSE(C, R) = (\frac{1}{|P|} \sum_{p \in P} (\mu_C(p) - \mu_R(p))^2)^{1/2}. \quad (2)$$

For a classification task involved with m ($m > 2$) classes, m fuzzy pattern trees will be constructed following the one-versus-rest decomposition such that each single fuzzy pattern tree is associated with a unique class. Whereas for binary classification problems, the predictions can be made with only one single fuzzy pattern tree via thresholding method [24]. The generation of a fuzzy pattern tree is an iterative process, which starts by initialising a candidate primitive tree with a single node representing the underlying target class C. A set of candidate trees CT^{t+1} is then generated by listing exhaustively through the possible combinations of available fuzzy propositions and aggregation functions. A local best tree $bestT$ is selected from the candidate set CT^t of current iteration such that $bestT = \arg\min_{ct \in CT^t} RMSE(C, R_{ct})$, where R_{ct} is the root of a candidate tree $ct, ct \in CT^t$.

The candidate tree in the next iteration $t+1$ is an expansion of the $bestT$ by replacing one of its leaf nodes with a subtree. The subtree is made up of three nodes including the original leaf node, a new leaf node selected among the set of unused fuzzy propositions, and an inner node selected among the set of predefined aggregation operators. The construction process of a fuzzy pattern tree can be briefly shown in Fig. 1, where the blue nodes are newly generated at the current iteration. Once all new candidates are generated, the tree with most gain as evaluated by (2) will be utilised in the next iteration. These steps are repeated until every leaf has reached a pre-defined maximum depth or the improvement of $RMSE(C, R_{bestT})$ is smaller than a pre-defined threshold. The detailed algorithm for constructing the pattern tree can be found in [23].

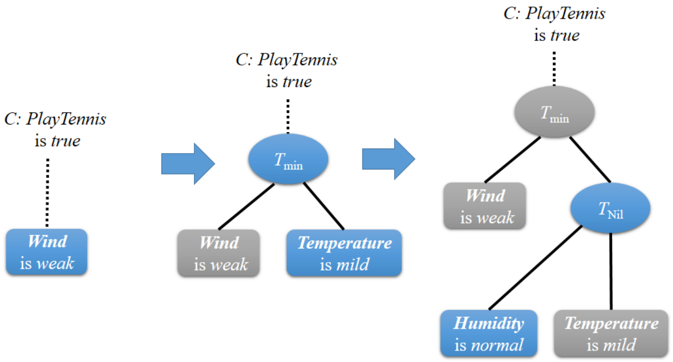

Fig. 1. An example of top-down construction of a fuzzy pattern tree. The tree on the left is a primitive tree. The tree in the middle consists of two leaves and the root. The right one consists of three leaves, one inner node, and the root.

B. Generating Preaggregation Functions for Fuzzy Pattern Trees

As reflected by the above construction process, the fuzzy pattern tree grows iteratively whereby the search space delimited by existing fuzzy proposition is refined by adding another proposition connected by aggregation functions. The aggregation functions that define the operations between fuzzy propositions, have a direct impact upon the space restricted by the propositions. As a result, the choices of aggregation functions to be applied during the construction of fuzzy pattern tree ultimately influence the overall performance of resultant tree.

It is worth noting that the very original fuzzy ID3 decision tree only employs a specific implementation of T-norm and T-conorm (i.e., $\min$ and $\max$) as logical operators for knowledge modelling. In fact, human decision makers do not correspond to the verbal and logical connectives 'and' and 'or' but almost always show some degrees of compensation when inferring a final decision from a set of criteria [25]. Therefore, fuzzy pattern trees also utilise more generic averaging operators such as the WA and OWA to cover the interval between the maximum and minimum of inputs. More recent fuzzy pattern trees in the literature have been developed to utilise the Choquet integral e.g., the generalisation of both the WA and OWA, as aggregators for inner nodes [24].

Inspired by the possible significance of aggregation functions in affecting the performance of fuzzy pattern tree, this paper proposes an initial work incorporating preggregation functions, which are deemed as a type of generalised aggregation functions in recent literature, to the construction of fuzzy pattern tree. Given that the WA and OWA are special cases of Choquet integral and have been successfully applied to fuzzy pattern trees' construction, a Choquet-integral-like method [19] of generating preaggregation functions is employed in this paper. The generated preaggregation functions work as the candidate operators to be selected as inner nodes or the root while the pattern tree expands.

The preaggregation functions are formed in a way similar to the Choquet integral, where the product operation in (1) is replaced by other operators. Formally, let $m : 2^N \to [0,1], N = \{1, \cdots, n\}$ be a fuzzy measure and $F : [0,1]^2 \to [0,1]$ be a function such that for all $x, y \in [0,1]$, it satisfies $F(x,y) \le x, F(x,1) = x, F(0,y) = 0$ and F is $(1,0)$-increasing. Then, an idempotent and averaging preaggregation function $P_m^F : [0,1]^n \to [0,n]$ can be defined by

$$P_m^F(x) = \sum_{i=1}^{n} F((x_{(i)} - x_{(i-1)}), m(A_{(i)})) \tag{3}$$

where $x_{(1)}, \cdots, x_{(n)}$ is an increasing permutation on input x, i.e., $0 \le x_{(1)} \le \cdots \le x_{(n)}$, with the convention that $x_{(0)} = 0$, and $A_{(i)} = \{(i), \cdots, (n)\}$ is the subset of indices of $n-i+1$ largest components of x.

In this paper, the T_{min}, T_{p}, $T_{\text{Ł}}$, T_{Nil}, and T_{Ham} are selected as the function F to define preaggregation functions by using the Choquet-integral-like method. Since fuzzy pattern trees are binary trees, i.e., each node only contains two children, all generated preaggregation functions are bivariate (with respect to inputs $x, y \in [0,1]$). For simplicity, only the bivariate form of those generated preaggregation functions are listed. The fuzzy measure m adopted here is the symmetric measure m_s^W. It is worth noting that since only two inputs are considered, $m_s^W(A_{(1)}) = 1$, $m_s^W(A_{(2)}) = 1 - w_1 = w_2$, $x_{(0)} = 0$, $x_{(1)} = \min\{x,y\}$, and $x_{(2)} = \max\{x,y\}$.

- The preaggregation function generated by T_{min} is:

$$P_{m_s^W}^{T_{\text{min}}}(x,y) = \min\{x,y\} + \min\{|x-y|, w_2\}.$$

- The preaggregation function generated by T_{p} is:

$$\begin{aligned} P_{m_s^W}^{T_{\text{p}}}(x,y) &= \min\{x,y\} + |x-y| \cdot w_2 \\ &= \min\{x,y\} \cdot w_1 + \max\{x,y\} \cdot w_2. \end{aligned}$$

By using the T_{p} norm, (3) is degenerated into (1) and $P_{m_s^W}^{T_{\text{p}}}$ is the OWA associated to (w_1, w_2).

- The preaggregation function generated by $T_{\text{Ł}}$ is:

$$P_{m_s^W}^{T_{\text{Ł}}}(x,y) = \min\{x,y\} + \max\{0, |x-y| - w_1\}.$$

- The preaggregation function generated by T_{Nil} is:

$$P_{m_s^W}^{T_{\text{Nil}}}(x,y) = \begin{cases} \min\{x,y\}, \text{ if } |x-y| \le w_1 \\ \min\{x,y\} + \min\{|x-y|, w_2\}, \text{ otherwise.} \end{cases}$$

Consider $w_1 = 0.3$, then $P_{m_s^W}^{T_{\text{Nil}}}(0.1, 0.5) = 0.5$ and $P_{m_s^W}^{T_{\text{Nil}}}(0.3, 0.5) = 0.3$. Therefore, $P_{m_s^W}^{T_{\text{Nil}}}$ is not an aggregation function.

- The preaggregation function generated by T_{Ham} is:

$$P_{m_s^W}^{T_{\text{Ham}}}(x,y) = \begin{cases} \min\{x,y\}, \text{ if } |x-y| = 0 \text{ and } w_2 = 0 \\ \min\{x,y\} + \dfrac{|x-y| \cdot w_2}{|x-y| + w_2 - |x-y| \cdot w_2} \\ \qquad, \text{ otherwise.} \end{cases}$$

The contours of resultant preaggregation functions are shown in Fig. 2, where the rows plot different T-norms and the columns plot correponding values of W. Similar to OWA operators, the behaviour [19] of preaggregation functions is also bounded in the range between $\min\{x,y\}$ and $\max\{x,y\}$.

IV. Experiment and Evaluation

In order to evaluate the performance of the proposed preaggregation-based fuzzy pattern tree, experiments are performed on six popular UCI benchmark data sets [26]. The full names of the data sets are listed in the leftmost column of Table I. For simplicity, linguistic terms for each data set are generated by fuzzifying each numeric feature into three fuzzy terms which indicate 'low', 'medium', and 'high', respectively. The membership functions follow the conventional triangular shapes and are defined as shown in Fig. 3.

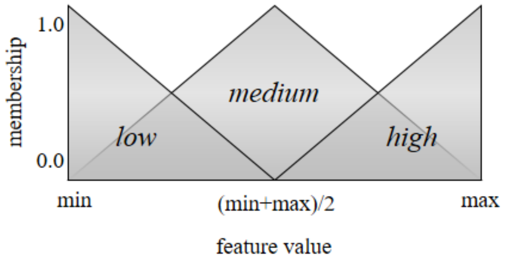

Fig. 3. Membership functions of linguistic terms. X-axis represents the feature value in the interval of [min, max] and Y-axis represents the membership degree in the unit interval $[0,1]$.

All the five types of preaggregation function listed in Fig. 2 are tested separately. For each type of preaggregation function, eleven different weighting vectors are employed to generate the candidate operators such that $w_1 = \{0.0, 0.1, \cdots, 1.0\}$. It is worth noting that the classic max and min operators can be derived from any of the five types of preaggregation functions by setting $w_1 = 0.0$ and $w_1 = 1.0$, respectively. Apart from the preaggregation functions, two generalised OWA operators [27] are also compared in the experiment and are defined as follows:

$$\text{OWA}^2(x,y) = (\min\{x,y\}^2 \cdot w_1 + \max\{x,y\}^2 \cdot w_2)^{\frac{1}{2}}$$

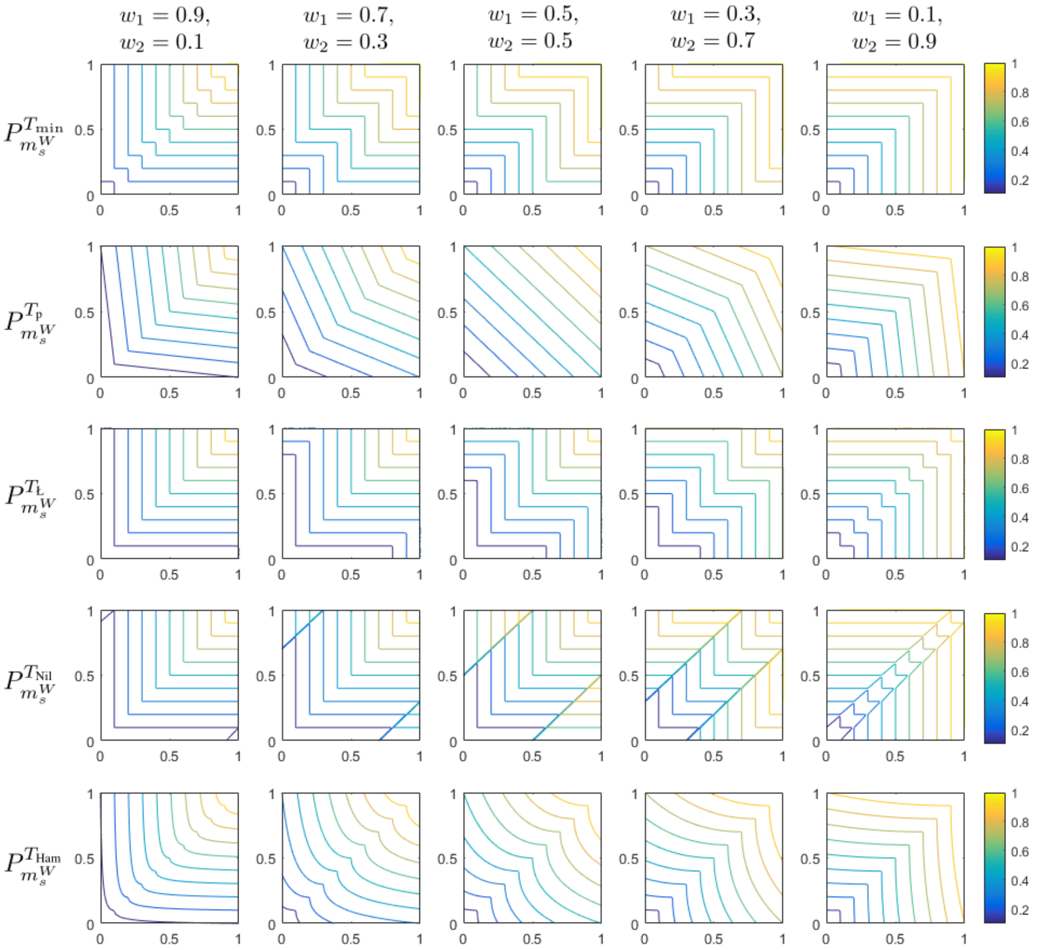

Fig. 2. Contours of generated preaggregation functions. X-axis represents the value of $x \in [0, 1]$ and Y-axis represents the value of $y \in [0, 1]$.

$$\mathrm{OWA}^{\frac{1}{2}}(x, y) = (\min\{x, y\}^{\frac{1}{2}} \cdot w_1 + \max\{x, y\}^{\frac{1}{2}} \cdot w_2)^2$$

The performance on accuracy is shown in Table I, which is obtained by the averaging 5 random runs of 10-fold cross-validation. The result with best performance on each data set is also highlighted in boldface in Table I.

It can be clearly seen from Table I that performances of the resulting fuzzy pattern trees incorporated with different aggregation functions differ from one another, which confirms the significance of preaggregation functions to the construction of fuzzy pattern trees. In particular, the $P^{T_{\mathrm{Nil}}}_{m_s^W}$ achieved the best accuracy on average supported with four best results out of six data sets. However this does not necessarily mean $P^{T_{\mathrm{Nil}}}_{m_s^W}$ always works best under all settings. It has been reported in recent literature [19] that $P^{T_{\min}}_{m_s^W}$ and $P^{T_{\mathrm{Ham}}}_{m_s^W}$ achieve better results than $P^{T_{\mathrm{Nil}}}_{m_s^W}$ when the generated preaggregation functions are applied to the fuzzy-rule-based classification system [28]. To benchmark the performance of proposed construction of fuzzy pattern trees, the results of classic decision tree (implemented in WEKA [29] as J48) are also reported in Table I. The numeric features in each data set are discretised into three bins for a fair comparison.

Take a closer look at the contours of the preaggregation functions as shown in Fig. 2. The contours of $P^{T_{\min}}_{m_s^W}$ and $P^{T_{\mathrm{Ham}}}_{m_s^W}$ are similar in the way that both are non-convex on two sides of symmetry axis where $x = y$. On the other hand, the contours of $P^{T_{\mathrm{Nil}}}_{m_s^W}$ can be generated as a combination of $P^{T_{\min}}_{m_s^W}$ and $P^{T_{\text{Ł}}}_{m_s^W}$, which is leveraged by a threshold. One possible reason of $P^{T_{\mathrm{Nil}}}_{m_s^W}$ outperforming $P^{T_{\min}}_{m_s^W}$ and $P^{T_{\mathrm{Ham}}}_{m_s^W}$ may be that the resultant fuzzy pattern trees only employ the bivariate form of preaggregation functions in this experiment, while $P^{T_{\min}}_{m_s^W}$ and $P^{T_{\mathrm{Ham}}}_{m_s^W}$ might be more powerful in taking multiple inputs [28]. Nevertheless, the experiment demonstrates the effectiveness of preaggregation functions in constructing fuzzy pattern trees, especially in comparison with the commonly used averaging operators such as OWA and generalised OWA.

V. Conclusion

This paper proposes an initial work to explore preaggregation functions with application to recently proposed fuzzy patter tree. The preaggregation functions generated by the Choquet-integral-like method and symmetric fuzzy measures are integrated into the construction of fuzzy pattern trees. The compared preaggregation functions are generated by five types

TABLE I
COMPARISON OF ACCURACY (%)

	$P^{T_{\min}}_{m^W_s}$	$P^{T_{\mathrm{p}}}_{m^W_s}$/OWA	$P^{T_{\mathrm{Ł}}}_{m^W_s}$	$P^{T_{\mathrm{Nil}}}_{m^W_s}$	$P^{T_{\mathrm{Ham}}}_{m^W_s}$	OWA^2	OWA$^{\frac{1}{2}}$	J48
Iris	85.87	80.8	81.6	**91.33**	81.6	78.8	82.93	96.80
Glass	60.16	62.72	57.72	**64.82**	62.84	59.26	62.55	62.39
Breast-cancer	72.46	70.56	72.32	**73.39**	69.58	71.34	71.06	74.63
Cleveland-heart-disease	80.8	81.79	80.68	82.19	81.12	80.47	**82.72**	75.97
Wine	91.56	91.69	**93.38**	92.35	91.23	91.24	91.12	89.88
Ecoli	74.1	74.59	71.66	**79.76**	76.07	72.87	75.18	78.29
Means	77.49	77.03	76.23	**80.64**	77.07	75.66	77.59	79.66

of T-norms, each of which is generated with eleven fixed weighting vectors. Experiments are performed in classification tasks on six benchmark data sets. The result indicates the $P^{T_{\mathrm{Nil}}}_{m^W_s}$ outperforms its counterparts and preaggregation functions can be more powerful than conventional OWA operators in constructing fuzzy pattern trees.

Whilst promising, the presented work also opens up an avenue for significant further investigation. For instance, many other preaggregation functions and weighting vectors [30] may also be applied. It would be useful to develop an algorithm for weighting vector optimisation in order to further enrich its performance. It is also interesting to use different types of preaggregation functions jointly rather than individually to build fuzzy pattern trees, which may be more powerful for knowledge modelling.

REFERENCES

[1] M. J. Cobo, M. Martínez, M. Gutiérrez-Salcedo, H. Fujita, and E. Herrera-Viedma, "25 years at knowledge-based systems: a bibliometric analysis," *Knowledge-Based Systems*, vol. 80, pp. 3–13, 2015.

[2] M. Nakano, A. Takahashi, and S. Takahashi, "Fuzzy logic-based portfolio selection with particle filtering and anomaly detection," *Knowledge-Based Systems*, vol. 131, pp. 113–124, 2017.

[3] T. Chen, Q. Shen, P. Su, C. Shang, "Induction of quantified fuzzy rules with Particle Swarm Optimisation," in *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).* IEEE, 2015, pp. 1–7.

[4] T. Chen, Q. Shen, P. Su, and C. Shang, "Fuzzy rule weight modification with particle swarm optimisation," *Soft Computing*, vol. 20, no. 8, pp. 2923–2937, 2016.

[5] T. Chen, P. Su, C. Shang, Q. Shen, "Reliability-guided fuzzy classifier ensemble," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).* IEEE, 2017, pp. 1–6.

[6] T. Chen, P. Su, C. Shang, and Q. Shen, "Weighted fuzzy rules optimised by particle swarm for network intrusion detection," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).* IEEE, 2018, pp. 1–7.

[7] Y. Yuan and M. J. Shaw, "Induction of fuzzy decision trees," *Fuzzy Sets and systems*, vol. 69, no. 2, pp. 125–139, 1995.

[8] A. Segatori, F. Marcelloni, and W. Pedrycz, "On distributed fuzzy decision trees for big data," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 1, pp. 174–192, 2018.

[9] P. Su, Q. Shen, T. Chen, and C. Shang, "Ordered weighted aggregation of fuzzy similarity relations and its application to detecting water treatment plant malfunction," *Engineering Applications of Artificial Intelligence*, vol. 66, pp. 17–29, 2017.

[10] Z. Huang, T. D. Gedeon, and M. Nikravesh, "Pattern Trees Induction: A New Machine Learning Method," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 4, pp. 958–970, 2008.

[11] R. Senge and E. Hüllermeier, "Pattern trees for regression and fuzzy systems modeling," in *2010 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).* IEEE, 2010, pp. 1–7.

[12] A. Shaker, R. Senge, and E. Hüllermeier, "Evolving fuzzy pattern trees for binary classification on data streams," in *Information Sciences*, vol. 220, pp. 34–45, 2013.

[13] G. Lucca, J. A. Sanz, G. P. Dimuro, B. Bedregal, H. Bustince, and R. Mesiar, "C_F-integrals: A new family of pre-aggregation functions with application to fuzzy rule-based classification systems," *Information Sciences*, vol. 435, pp. 94–110, 2018.

[14] G. Lucca, J. A. Sanz, H. Bustince, G. P. Dimuro, V. Gomes, R. C. C. Madureira, and P. Melo-Pinto, "Applying aggregation and preaggregation functions in the classification of grape berries," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).* IEEE, 2018, pp. 1–6.

[15] B. Schweizer and A. Sklar, *Probabilistic metric spaces.* Courier Corporation, 2011.

[16] R. R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988.

[17] T. Wilkin and G. Beliakov, "Weakly monotonic averaging functions," *International Journal of Intelligent Systems*, vol. 30, no. 2, pp. 144–169, 2015.

[18] H. Bustince, J. Fernández, A. Kolesárová, and R. Mesiar, "Directional monotonicity of fusion functions," *European Journal of Operational Research*, vol. 244, no. 1, pp. 300–308, 2015.

[19] G. Lucca, J. A. Sanz, G. P. Dimuro, B. Bedregal, R. Mesiar, A. Kolesárová, and H. Bustince, "Preaggregation Functions: Construction and an Application," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 2, pp. 260–272, 2016.

[20] J. Fodor, J. Marichal, M. Roubens, "Characterization of the ordered weighted averaging operators," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 2, pp. 236–240, 1995.

[21] T. Chen, C. Shang, P. Su, and Q. Shen, "Induction of accurate and interpretable fuzzy rules from preliminary crisp representation," *Knowledge-Based Systems*, vol. 146, pp. 152–166, 2018.

[22] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[23] R. Senge and E. Hüllermeier, "Top-down induction of fuzzy pattern trees," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 241–252, 2011.

[24] ——, "Fast Fuzzy Pattern Tree Learning for Classification," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 6, pp. 2024–2033, 2015.

[25] H.-J. Zimmermann and P. Zysno, "Latent connectives in human decision making," *Fuzzy sets and systems*, vol. 4, no. 1, pp. 37–51, 1980.

[26] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[27] R. R. Yager, "Generalized owa aggregation operators," *Fuzzy Optimization and Decision Making*, vol. 3, no. 1, pp. 93–107, 2004.

[28] E. Barrenechea, H. Bustince, J. Fernandez, D. Paternain, and J. A. Sanz, "Using the Choquet integral in the fuzzy reasoning method of fuzzy rule-based classification systems," *Axioms*, vol. 2, no. 2, pp. 208–223, 2013. [Online]. Available: http://www.mdpi.com/2075-1680/2/2/208

[29] E. Frank, M. A. Hall, and I. H. Witten, *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*, Morgan Kaufmann, Fourth Edition, 2016.

[30] P. Su, C. Shang, T. Chen, and Q. Shen, "Exploiting data reliability and fuzzy clustering for journal ranking," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1306–1319, 2017.