Weighted Pattern Trees: A Case Study with Customer Satisfaction Dataset

Zhiheng Huang[1], Masoud Nikravesh[1], Ben Azvine[2], and Tamás D. Gedeon[3]

[1] Electrical Engineering and Computer Science,
University of California at Berkeley, CA 94720, USA
{zhiheng,nikravesh}@cs.berkeley.edu
[2] Computational Intelligence Research Group, Intelligent Systems Research Center
BT Group Chief Technology Office, British Telecom
ben.azvine@bt.com
[3] Department of Computer Science,
The Australian National University, Canberra, ACT 0200, Australia
tom@cs.anu.edu.au

Abstract. A pattern tree [1] is a tree which propagates fuzzy terms using different fuzzy aggregations. Each pattern tree represents a structure for an output class in the sense that how the fuzzy terms aggregate to predict such a class. Unlike decision trees, pattern trees explicitly make use of t-norms (i.e., AND) and t-conorms (OR) to build trees, which is essential for applications requiring rules connected with t-conorms explicitly. Pattern trees can not only obtain high accuracy rates in classification applications, but also be robust to over-fitting. This paper further extends pattern trees approach by assigning certain weights to different trees, to reflect the nature that different trees may have different confidences. The concept of weighted pattern trees is important as it offers an option to trade off the complexity and performance of trees. In addition, it enhances the semantic meaning of pattern trees. The experiments on British Telecom (BT) customer satisfaction dataset show that weighted pattern trees can slightly outperform pattern trees, and both of them are slightly better than fuzzy decision trees in terms of prediction accuracy. In addition, the experiments show that (weighted) pattern trees are robust to over-fitting. Finally, a limitation of pattern trees as revealed via BT dataset analysis is discussed and the research direction is outlined.

1 Introduction

Most of the existing fuzzy rule induction methods including fuzzy decision trees [9] (the extension of the classic decision tree induction method by Quinlan [6]) focus on searching for rules which only use t-norm operators [7] such as the MIN and algebraic MIN. Disregarding of the t-conorms such as MAX and algebraic MAX is due to the fact that any rule using t-conorms can be represented by several rules which use t-norms only. This is certainly true and it is helpful to simplify the rule induction process by considering t-norms only. However, it may fail to generate important rules in which fuzzy terms are explicitly connected with t-conorms. Research has been conducted to resolve this problem. For example, Kóczy, Vámos and Biró [3] have proposed fuzzy signatures to model the complex structures of data points using different aggregation operators including MIN, MAX, and average etc. Mendis, Gedeon and Kóczy [4] have investigated different aggregations in fuzzy signatures. Nikravesh [5] has presented evolutionary computation (EC) based multiple aggregator fuzzy decision trees.

Huang and Gedeon [1] have first introduced the concept of pattern trees and proposed a novel pattern tree induction method by means of similarity measures and different aggregations. This paper extends that work to assign certain weights to different pattern trees. As a result, it enhances the semantic meaning of pattern trees and makes them more comprehensible for users. The experiments on BT customer satisfaction dataset show that weighted pattern trees can slightly outperform pattern trees. In addition, this paper shows that pattern trees and weighted pattern trees perform more consistently than fuzzy decision trees. The former are capable of generating classifiers with good generality, while the latter can easily fall into the trap of over-fitting. In fact, weighted pattern trees with only two or three tree levels (depth of tree) are good enough for most experiments carried out in this paper. This provides a very transparent way to model real world applications.

The rest of the paper is arranged as follows: Section 2 provides the definitions for similarity, aggregations and pattern trees, and briefly outlines the pattern tree induction method. Readers may refer to [1][2] for detailed discussion. Section 3 suggests the concept of weighted pattern trees and shows how to use them for classification. Section 4 presents the experimental results over BT customer satisfaction dataset. Finally, Section 5 concludes the paper and points out further research work.

2 Definitions and Pattern Tree Induction

Let A and B be two fuzzy sets [10] defined on the universe of discourse U. The root mean square error (RMSE) of fuzzy sets A and B can be computed as

$$RMSE(A,B) = \sqrt{\frac{\sum_{j=1}^{m}(\mu_A(x_j) - \mu_B(x_j))^2}{m}}, \tag{1}$$

where x_j, $j = 1, \ldots, m$, are the crisp values discretized in the variable domain, and $\mu_A(x_j)$ and $\mu_B(x_j)$ are the fuzzy membership values of x_j for A and B. The RMSE based fuzzy set similarity can thus be defined as

$$S(A,B) = 1 - RMSE(A,B). \tag{2}$$

The larger the value of $S(A,B)$, the more similar A and B are. As $\mu_A(x_j), \mu_B(x_j) \in [0,1]$, $0 \le S(A,B) \le 1$ holds according to (1) and (2). Note that the pattern tree induction follows the same principle if alternative fuzzy set similarity definitions such as Jaccard are used.

Fuzzy aggregations are logic operators applied to fuzzy membership values or fuzzy sets. They have three sub-categories, namely t-norm, t-conorm, and averaging operators such as weighted averaging (WA) and ordered weighted averaging (OWA) [8].

Triangular norms were introduced by Schweizer and Sklar [7] to model distances in probabilistic metric spaces. In fuzzy sets theory, triangular norms (t-norm) and triangular conorms (t-conorm) are extensively used to model logical operators *and* and *or*. The basic t-norm and t-conorm pairs which operate on two fuzzy membership values a and b, $a, b \in [0, 1]$ are shown in Table 1. Although

Table 1. Basic t-norms and t-conorms pairs

Name	t-norm	t-conorm
MIN/MAX	$min\{a, b\} = a \wedge b$	$max\{a, b\} = a \vee b$
Algebraic AND/OR	ab	$a + b - ab$
Łukasiewicz	$max\{a + b - 1, 0\}$	$min\{a + b, 1\}$
EINSTEIN	$\frac{ab}{2-(a+b-ab)}$	$\frac{a+b}{1+ab}$

the aggregations shown only apply to a pair of fuzzy values, they can apply to multiple fuzzy values as they retain associativity. The definition of WA and OWA are shown as follows:

Definition 1. *A WA operator of dimension n is a mapping $E : \mathbb{R}^n \rightarrow \mathbb{R}$, that has an associated n-elements vector $w = (w_1, w_2, \ldots, w_n)^T$, $w_i \in [0, 1]$, $1 \leq i \leq n$, and $\sum_{i=1}^{n} w_i = 1$ so that $E(a_1, \ldots, a_n) = \sum_{j=1}^{n} w_j a_j$.*

Definition 2. *An OWA operator [8] of dimension n is a mapping $F : \mathbb{R}^n \rightarrow \mathbb{R}$, that has an associated n-elements vector $w = (w_1, w_2, \ldots, w_n)^T$, $w_i \in [0, 1]$, $1 \leq i \leq n$, and $\sum_{i=1}^{n} w_i = 1$ so that $F(a_1, \ldots, a_n) = \sum_{j=1}^{n} w_j b_j$, where b_j is the jth largest element of the collection $\{a_1, \ldots, a_n\}$.*

A fundamental difference of OWA from WA aggregation is that the former does not have a particular weight w_i associated for an element, rather a weight is associated with a particular ordered position of the element.

A pattern tree is a tree which propagates fuzzy terms using different fuzzy aggregations. Each pattern tree represents a structure for an output class in the sense that how the fuzzy terms aggregate to predict such a class. The output class is located at the top as the root of this tree. The fuzzy terms of input variables are on different levels (except the top) of the tree. They use fuzzy aggregations to aggregate from the bottom to the top (root). Assume two fuzzy variables A and B each have two fuzzy linguistic terms A_i and B_i, $i = \{1, 2\}$, and the task is to classify the data samples to either class X or Y. Fig. 1 shows two example pattern trees, with one for class X and the other for Y. It can be seen that pattern trees are built via the aggregation of fuzzy terms. For example, the pattern tree for X is equivalent to fuzzy rule $(B_1 \wedge A_2) \vee A_1 \Rightarrow X$.

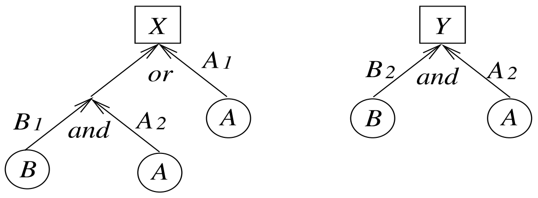

Fig. 1. Two example pattern trees

For a classification application which involves several output classes, the worked model should have as many pattern trees as the number of output classes, with each pattern tree representing one class. When a new data sample is tested over a pattern tree, it traverses from the bottom to the top and finishes with a truth value, indicating a degree to which this data sample belongs to the output class of this pattern tree. The output class with the maximal truth value is chosen as the prediction class. For example, consider that a fuzzy data $A_1 = 0.8$, $A_2 = 0.2$, $B_1 = 0$, and $B_2 = 1$ is given for classification. As the truth values of this data over pattern trees for class X and Y are 0.8 and 0.2 respectively, X is chosen as the output class.

The pattern tree induction method as proposed in [1][2] is briefly outlined as follows. Readers may refer to [1][2] for detailed discussion. Without losing generality, assume a dataset has n input variables A_i, $i = 1, 2, \ldots, n$ and one output variable B. Further assume that input variables have m fuzzy linguistic terms denoted as A_{ij}, $i = 1, 2, \ldots, n$, and $j = 1, 2, \ldots, m$, and output variable has k fuzzy or linguistic terms denoted as B_j, $j = 1, 2, \ldots, k$. That is, each data point is represented by a fuzzy membership value vector of dimension $(nm + k)$. The task is to build k pattern trees for the k output classes (fuzzy or linguistic terms).

The process of building a pattern tree, say for class B_0, is described as follows:

1. From fuzzy term set $\mathcal{S} = \{A_{ij}\}$, $i = 1, 2, \ldots, n$, and $j = 1, 2, \ldots, m$, choose a fuzzy linguistic term $A_{i'j'} \in \mathcal{S}$, which has the highest similarity to the output class B_0 as the initial tree. The fuzzy term set is updated as $\mathcal{S} = \mathcal{S} - A_{i'j'}$. The exclusion of fuzzy term $A_{i'j'}$ from $\mathcal{S}$ is to prevent $A_{i'j'}$ from being used more than once in the tree.
2. Try aggregating the current tree with all fuzzy linguistic terms at set $\mathcal{S}$ in turn with different aggregations. Grow the current tree using the term $A_{i'j'}$ from set $\mathcal{S}$ and aggregation which together lead to the highest similarity. The fuzzy term set is updated as $\mathcal{S} = \mathcal{S} - A_{i'j'}$.
3. Keep applying 2 until no fuzzy term and aggregation lead to a higher similarity than the current one.

The above actually presents the induction for *simple pattern trees*. Its extension, the *general pattern trees* induction [2], considers to aggregate not only fuzzy terms, but also other pattern trees. In general, simple pattern trees not only produce high prediction accuracy, but also preserve compact tree structures, while general pattern trees can produce even better accuracy, but as a compromise produce more complex tree structures. Subject to the particular demands (comprehensibility or performance), simple pattern trees and general pattern trees provide an highly effective methodology for real world applications.

3 Weighted Pattern Trees

The classification using pattern trees discussed in section 2 is based on the assumption that all pattern trees each have the same confidence on predicting a particular class, though it is not always the case in real world applications. *Weighted trees* are introduced to resolve this problem. For each tree, the similarity of such tree to the output class is served as a degree of confidence, to reflect how confident to use this tree to predict such a class. For example, if the two trees in Fig. 1 have similarities of 0.1 and 0.8 respectively, they can be called weighted pattern trees with weights of 0.1 and 0.8. The prediction using weighted pattern trees is the same as that using pattern trees, except that the final truth values are multiplied by the weights of trees. As an example, let's revise the classification problem in section 2; consider classifying the fuzzy data $A_1 = 0.8$, $A_2 = 0.2$, $B_1 = 0$, and $B_2 = 1$ over pattern trees (with weights of 0.1 and 0.8) in Fig. 1, its truth values over pattern trees for class X and Y change to 0.08 and 0.16 respectively, and Y (rather than X) is therefore chosen as the output class. This reflects the fact that, if a tree has a low weight, even an input data has a high firing strength over such pattern tree, the prediction is not confident. Note that this example is merely used to show how weighted pattern trees work. In practice, a pattern tree with weight of 0.1 may not be trusted to predict a class.

The concept of weighted pattern trees is important. It offers an option to trade off the complexity and performance of pattern trees. The pattern tree building process can stop at very compact trees, if it detects that the similarities (weights) of such trees are already larger than a user pre-defined threshold. In addition, it enhances the comprehensibility of pattern trees. For example consider the construction of the pattern tree for class Y in Fig. 1, assume that the tree growing from the primitive tree $B_2 \Rightarrow Y$ to $B_2 \wedge A_2 \Rightarrow Y$ leads to the weight increase from 0.6 to 0.8, this gradual change can be interpreted in a comprehensible way:

$$\text{IF } B = B_2 \text{ THEN it is } \textbf{possible} \text{ that } \mathit{class} = Y, \tag{3}$$

$$\text{IF } B = B_2 \text{ AND } A = A_2 \text{ THEN it is } \textbf{very possible} \text{ that } \mathit{class} = Y, \tag{4}$$

if users pre-define semantic ranges of weights, say *less possible*: $[0, 0.3)$, *possible*: $[0.3, 0.7)$, and *very possible*: $[0.7, 1]$. Thus, the graduate change of confidence of pattern trees can be monitored from the pattern tree induction process. This provides a very transparent way for fuzzy modeling.

4 Experimental Results

In this section, different variants of pattern trees, namely simple pattern trees, weighted simple pattern trees, pattern trees, and weighted pattern trees, are applied to a sample customer satisfaction dataset from BT. This dataset has a total of 26 input parameters representing ease of contact, problem understanding, service quality, repair time, and overall event handling. Among the input parameters, 6 are numerical parameters and the rest 20 are category ones, with the number of possible values being from 2 up to 17. The output parameter consists of 7 classes reflecting varying degrees of customer satisfaction.

The BT customer satisfaction dataset has 16698 data points in total. Let *ds*, *ds-odd* and *ds-even* be the datasets which contain the whole, the odd numbered, and the even numbered data points respectively. The number of data per class for these three datasets are shown in Table 2, with ci, $i = 0, \ldots, 6$ standing for class i. As can be seen, this dataset is not well balanced as the number of data

Table 2. Number of data per class for ds, ds-odd and ds-even datasets

	c0	c1	c2	c3	c4	c5	c6
ds	1895	7289	4027	382	1361	853	891
ds-odd	949	3659	1990	197	660	448	446
ds-even	946	3630	2037	185	701	405	445

per class varies significantly. The experiments of (weighted) pattern trees are carried out in three combinations of training-test datasets, namely, *odd-even, even-odd*, and *ds-ds*. In all experiments, a simple fuzzification method based on three evenly distributed trapezoidal membership functions for each numerical input parameter is used to transform the crisp values into fuzzy values. All aggregations as listed in Table 1 are allowed and the similarity measure as shown in (2) is used.

4.1 Prediction Accuracy and Overfitting

The prediction accuracy and rule number of the fuzzy decision trees (FDT) with respect to the number of data points per leaf node (used as criteria to terminate the training), over different combinations of training-test sets are shown in Fig. 2. It reveals that in general the larger number of data points per leaf node, the more compact of the decision trees would be, thus leading to more general trees. The prediction accuracy of pattern trees (PT) and weighted pattern trees (WPT) with respect to different tree levels, over different combinations of training-test sets is shown in Fig. 3. It reveals that (simple) pattern trees maintain good generality even their structure becomes complex.

The experiments show that weighted pattern trees and pattern trees perform roughly the same. In fact, the former slightly outperform the latter. Table 3 shows the highest prediction accuracy of fuzzy decision trees, (weighted) simple pattern trees and (weighted) pattern trees over different combinations of

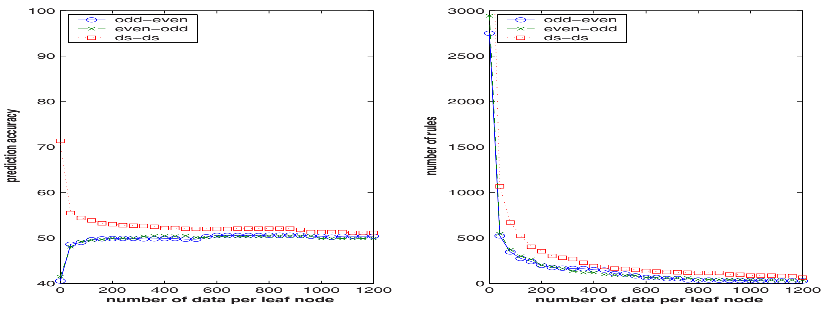

Fig. 2. Prediction accuracy and rule number of fuzzy decision trees with different number of data points per leaf node

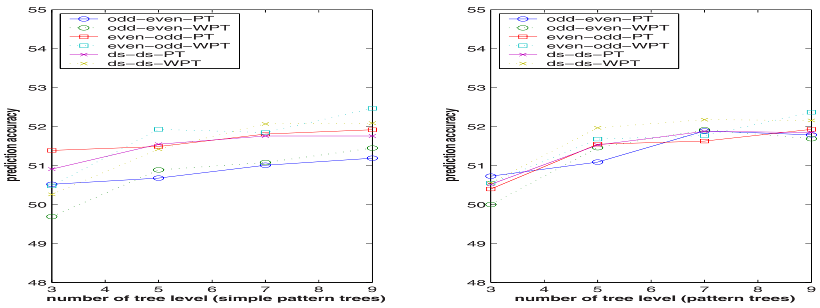

Fig. 3. Prediction accuracy of pattern trees and weighted pattern trees with different tree levels

Table 3. Highest prediction accuracy of fuzzy decision tree, pattern trees, and weighted pattern trees

	FDT	SimPT		PT	
		no weight	weight	no weight	weight
odd-even	50.62%	51.19%	51.45%	51.89%	51.92%
even-odd	50.47%	51.92%	52.47%	51.93%	52.37%
ds-ds	71.36%	51.82%	52.09%	51.88%	52.18%

training-test sets. Both weighted and un-weighed pattern trees can obtain higher prediction accuracy than fuzzy decision trees in *odd-even* and *even-odd* combinations. However, if considering *ds-ds* combination, fuzzy decision trees perform much better. This just reflects the overfitting of fuzzy decision trees, since fuzzy decision trees generate large differences in classification accuracy between the *odd-even*, *even-odd* combinations and *ds-ds* one. The reason is that decision tree induction considers only a portion of the whole training dataset in choosing the branches at low levels of trees. The lack of using the whole training dataset inevitably prevents the method finding generalized tree structures for all the dataset. In contrast, pattern trees make use of the whole data in building each level of the tree, which ensures the tree to keep good generality for classifications. Therefore, even complex pattern trees do not suffer from over-fitting.

In addition, the experiments show that (weighted) pattern trees tend to converge to a accuracy rate when the number of tree level becomes large. It has no trend of overfitting. This property is essential to ensure a stable, compact and effective fuzzy model for the problem at hand. In fact, (weighted) pattern trees with two or three level perform very well for all conducted experiments. That means, pattern trees which consist of maximal $2^3 = 8$ leaf nodes can perform well, in contrast to tens, or even hundred rules used in fuzzy decision trees. This provides a superb solution to achieve a highly effective as well as compact fuzzy model.

4.2 Approximate Accuracy

Section 4.1 presented the prediction accuracy of trees in a very strict way. That is, if and only if a data is predicted exactly as its class, this prediction is counted as a correct one. In other words, there is no distinction between "close" errors and "gross" errors. In BT customer dataset, this distinction is necessary as it reflects how far the prediction is away from the actual class. It is much worse if a data of class 0 is mis-predicted to class 5 rather than to class 1. To resolve this problem, three accuracy estimations, namely *accuracy 1*, *accuracy 2*, and *accuracy 3* are employed to estimate prediction accuracy which has no tolerance (the same as the one used in Section 4.1), tolerance of adjacent mis-prediction, and tolerance of mis-prediction within two closest neighbor classes in either direction, respectively. For example in the BT dataset, the mis-prediction of a class 0 data to class 2 is still counted as a correct prediction in the estimation of *accuracy 3*, although it is not counted in either *accuracy 1* or *accuracy 2*.

Table 4 shows the highest prediction accuracy of fuzzy decision trees, (weighted) simple pattern trees and (weighted) pattern trees over odd-even combination of training-test sets (the results on even-odd and ds-ds combinations are similar and thus omitted). Both weighted and unweighted pattern trees can obtain higher prediction accuracy than fuzzy decision trees in estimation of accuracy 1 and 2. In estimation of accuracy 3, weighted pattern trees perform the best, and fuzzy decision trees outperform unweighted pattern trees. Generally, accuracy 2

Table 4. Highest prediction accuracy of fuzzy decision trees, pattern trees, and weighted pattern trees over odd-even training-test combination

	FDT	SimPT		PT	
		no weight	weight	no weight	weight
accuracy 1	50.62%	51.19%	51.45%	51.89%	51.92%
accuracy 2	84.02%	84.08%	84.68%	84.44%	84.82%
accuracy 3	92.13%	91.74%	92.70%	91.85%	92.29%

and 3 are consistent with accuracy 1. Pattern trees with a high value of accuracy 1 usually have high values of accuracy 2 and 3. This table also shows that both fuzzy decision trees and pattern trees can obtain over 80% prediction accuracy if the closest error can be tolerated.

4.3 Interpretation of Pattern Trees

Each pattern tree can be interpreted as a general rule. Considering building level 5 simple pattern trees using odd dataset, 7 simple pattern trees can be obtained, with each representing one output class. Fig. 4 shows the tree for class 0. The ellipses are the input parameters and the rectangle is the output class 0. Over each branch, i and Fi, $i = 0, \ldots$, are category values and fuzzy terms associated with each input parameter. All aggregators as shown in Table 1 are allowed to be used in pattern trees. For example, *A_AND* is algebraic AND, and *WA_0.84* is weighted average with weight vector $w = (0.84, 0.16)$.

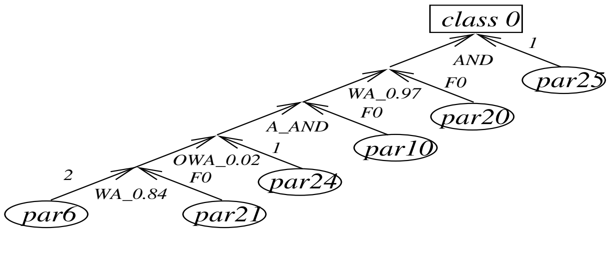

Fig. 4. Pattern tree for class 0 using *odd* dataset

Fig. 4 roughly indicates that one example combination yielding highly satisfied customers are: no call re-routing, fast fault reporting time, high technician competence, being well-informed through the repair process, and high satisfaction with company/product in general. Here, we say roughly, as we use different aggregations such as weighted average (WA), ordered weighted average (OWA), algebraic and (A_AND) etc. rather than simple AND.

These 7 pattern trees obtains an accuracy of 51.46%. In particular, the confusion table is shown in Table 5, where SA and SP are number of data for actual and predicted classes respectively.

4.4 Limitation

It is a little strange that no prediction is made to *c0* for all test data. From table 5, it can be seen that nearly all data (884 out of 946 in fact) with class 0 are mis-classified to class 1. A first intuition is to raise the weight of pattern tree for class 0. However, this does not work; the raise does not only lead to the data of class 0 to be classified correctly, but also lead to the majority of data of class 1 to be classified as class 0. Considering that there are 3630 data of class 1 and

Table 5. Confusion table for pattern tree prediction using odd-even combination

		Prediction							
		c0	c1	c2	c3	c4	c5	c6	SA
Actual	c0	0	884	51	0	3	2	6	946
	c1	0	3283	309	0	10	15	13	3630
	c2	0	1122	751	1	53	72	38	2037
	c3	0	59	94	0	6	20	6	185
	c4	0	122	395	1	30	104	49	701
	c5	0	50	142	0	23	120	70	405
	c6	0	35	129	0	37	131	113	445
	SP	0	5555	1871	2	162	464	295	8349

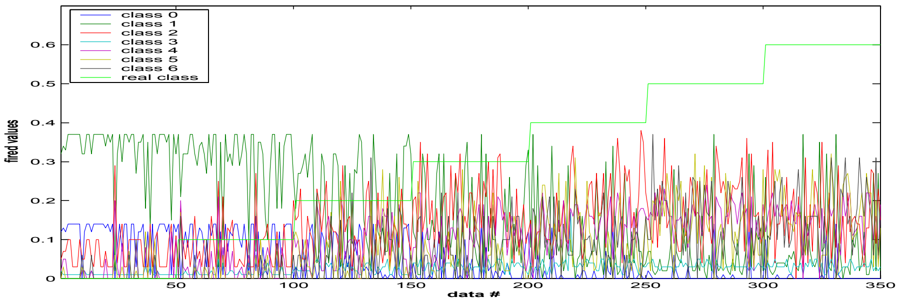

Fig. 5. Fired values of first 50 data points per class in *even* dataset over pattern trees constructed from *odd* dataset

only 946 data of class 0 in even dataset, the raise of weight for class 0 tree would therefore cause more mis-classifications. This can be seen in Fig. 5, where the fired values of first 50 data points per class in *even* dataset over pattern trees constructed from *odd* dataset are shown. The real class line indicates the real classes of the data; for example, data numbered from 0 to 49 have class 0, and those from 50 to 99 have class 1.

The phenomena of no prediction on particular classes also occurs in fuzzy decision trees. Considering the highest accuracy of 50.62% which fuzzy decision trees can obtain over odd-even combination, no data is predicted to *c3*, *c4* or *c5*, due to the small fraction of data points in those classes.

Table 6. Confusion table for prediction of both fuzzy decision trees and pattern trees using new training and test datasets

		Prediction		
		c0	c1	SA
Actual	c0	0	884	884
	c1	1	3282	3283
	SP	1	4166	4167

An interesting experiment is carried out trying to improve the prediction accuracy for class 0 in Table 5. The data of classes 0 and 1 in odd dataset are selected as a new training dataset, and the data which are of classes 0 and 1 in even dataset and are classified as class 1 in Table 5 are selected as a new test dataset. Both fuzzy decision trees and pattern trees are applied to the new training data and tested over the new test data. Surprisingly, they obtain the same highest accuracy of 78.76%. Table 6 shows the confusion table, which only has one data predicted as class 0 (and it is wrong actually). It can be concluded that the data of class 0 and class 1 can not be separated properly by either fuzzy decision trees or pattern trees.

5 Conclusions

This paper further extends pattern trees approach by assigning certain weights to different trees. The concept of weighted pattern trees is important as it not only offers an option to trade off the complexity and performance of trees, but also enhances the semantics of pattern trees.

The experiments on British Telecom customer satisfaction dataset show that weighted pattern trees can slightly outperform pattern trees, and both of them are slightly better than fuzzy decision trees in terms of prediction accuracy. In addition, the experiments show that (weighted) pattern trees are robust to over-fitting. In practice, weighted pattern trees with only two or three tree levels are good enough for most experiments carried out in this paper. This of course provides a very transparent way to model the problems at hand.

Further research on assignment of weights to pattern trees is necessary. The current version simply makes use of similarity measures as weights. More sophisticated assignment may be more suitable and can therefore lead to higher accuracy.

Acknowledgment

The authors would like to thank Marcus Thint for his helpful discussions on the work presented here.

References

1. Huang, Z. H. and Gedeon, T. D., "Pattern trees," *IEEE International Conference on Fuzzy Systems*, pp. 1784 - 1791, 2006.
2. Huang, Z. H., Gedeon, T. D. and Nikravesh, M., "Pattern trees," submitted to *IEEE Transactions on Fuzzy Systems*, 2006.
3. Kóczy, L. T., Vámos, T. and Biró, G., "Fuzzy signatures," *EUROFUSE-SIC*, pp. 210 – 217, 1999.
4. Mendis, B. S. U., Gedeon T. D., and Kóczy, L. T., "Investigation of aggregation in fuzzy signatures," *3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems*, 2005.

5. Nikravesh, M., “Soft computing for perception-based decision processing and analysis: web-based BISC-DSS,” *Studies in Fuzziness and Soft Computing*, vol. 164, pp. 93 -188, Springer Berlin/Heidelberg, 2005.
6. Quinlan, J. R., “Decision trees and decision making,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 339 – 346, 1994.
7. Schweizer, B. and Sklar, A., “Associative functions and abstract semigroups,” *Publ. Math. Debrecen*, vol. 10, pp. 69 – 81, 1963.
8. Yager, R. R., “On ordered weighted averaging aggregation operators in multicritera decison making,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 18, pp. 183 – 190, 1988.
9. Yuan, Y. and M. J. Shaw, “Induction of fuzzy decision trees,” *Fuzzy Sets and Systems*, vol. 69, no. 2, pp. 125 – 139, 1995.
10. Zadeh, L. A., “Fuzzy sets,” *Information and Control*, vol. 8, pp. 338 – 353, 1965.