DESIGN OF NEURO-FUZZY DECISION TREES

Tatyana Abramova[a]

National Research Tomsk State University, 634050, Tomsk, Russia

Abstract. In order to improve accuracy of fuzzy decision trees classification we propose a procedure of parameters adaptation by means of neural network training. In the direct cycle, fuzzy decision trees are based on the algorithm of fuzzy ID3; in the feedback cycle, parameters of fuzzy decision trees are adapted based on stochastic gradient algorithm by traverse to the root nodes back from the leaves. Using this strategy, the hierarchical structure of the fuzzy decision trees remains fixed.

1 Introduction

Domestic and foreign literature describes the use of decision trees as a powerful evolutionary methodology for solving problems of classification and regression [1–5]. Being a DATA MINING tool (detection of hidden knowledge from data), decision trees are used for search and retrieval of interpretable classification rules, which are clear for humans. We should note that many packages for intellectual data analysis already contain methods for constructing decision trees; they are the perfect tool for decision support systems.

2 Principles of constructing decision trees

Decision tree is a tree, the leaves of which contain values of the target function, and the remaining nodes contain branch conditions, which determine the edge to go (for example, "Sex is male"). If the condition is true for this observation, transition is made along the left edge, if it is false – along the right. Usually, each node includes checking of one independent variable. Sometimes two independent variables are compared with each other in the tree node, or a function of one or more variables is determined.

If a variable that is checked in the node takes categorial values, then the branch emanating from the tree node corresponds to each possible value. If the value of the variable is a number, it is checked to see if the value is greater or less than a constant. Sometimes, numerical range is divided into intervals (to check if the value hits one of the intervals).

The leaves of the trees correspond to the values of the dependent variable, i.e. classes. Figure 1 shows the iris classification tree. Classification contains three classes (marked with red, blue and green on the Figure 1) and has the parameters: length \ width of sepals (SepalLen, SepalWid) and length \ width of petals (PetalLen, PetalWid).

[a] Corresponding author: tanusha-atv@mail.ru

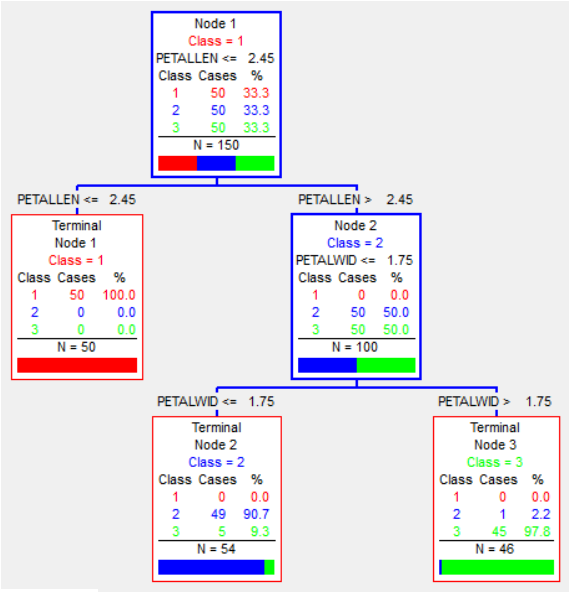

Figure1. Iris classification tree.

As we can see, each node contains class belonging (depending on the fact, what elements have hit this node in a greater number), number of observations N, and number of each class. Not leafy tops also contain a transition condition – to one of the subsidiaries. Sample is divided according to these conditions. As a result, the tree has classified initial data (exactly initial data, those on which it was trained) almost perfectly (6 out of 150 are not correct).

3 The main methods, which use decision trees

Classification and regression trees (CART) was the first method invented in 1983 by four well-known scientists in the field of data analysis: Leo Breiman, Jerome Friedman, Richard Olshen and Stone (Table 1) [2].

The essence of the algorithm is in ordinary construction of a decision tree [6]. On the first iteration, we build all possible (in a discrete sense) hyper planes, which would divide our space into two. Number of observations in each sub-space of different classes is counted for each subdividing. As a result, we select subdividing, which has maximally allocated observation of one of the classes in one of the sub-spaces. Accordingly, this subdividing is the root of our decision tree, and two subdivisions will be the leaves on this iteration.

On the next iterations, we take the worst leave (in the sense of number of observations of different classes) and conduct the same operation on its subdividing. As a result, the leave becomes a node with some subdividing and two leaves.

We continue to do so until we reach the limit on the number of nodes, or the overall error (the number of misclassified observations over the tree) fails to improve from one iteration to another. However, the resulting tree will be "retrained" (will be made-to-the training sample) and, accordingly, will not give normal results on other data. In order to avoid "retraining", it is possible to use test samples (or cross-validation) and, accordingly, to make reverse analysis (so-called, pruning), when the tree is reduced depending on the result on the test sample [7].

This is a relatively simple algorithm, which results in one decision tree. It is convenient for the primary data analysis, for example, to check presence of relationships between variables.

Random Forest is a method invented after CART by one of the four scientists - Leo Breiman in co-authorship with Adele Cutler [3]. The method is based on the use of committee (ensemble) of decision trees.

The essence of the algorithm is that random sampling of variables is made on each iteration, and then, constructing decision trees starts on this new sample. Besides, "bagging" takes place – sampling of random two-thirds of observations for training, and the remaining one-third is used to evaluate the results. This operation is done hundreds or thousands times. The resulting model will be the result of "voting" of a tree set obtained during simulation.

Stochastic Gradient Boosting is a data analysis method introduced by Jerome Friedman [4] in 1999. It represents solution of the regression problem (which can include classification) by constructing committee (ensemble) of "weak" predictive decision trees.

On the first iteration, a decision tree limited by number of nodes is constructed. Then the difference is counted between the value predicted by the resulting tree and multiplied by learnrate ("weakness" coefficient of each tree) and the unknown variable on this step. The next iteration is based on this difference. This continues until the result improves. It means that on every step we try to correct the mistakes of the previous tree. However, it is better to use check data (not involved in the simulation), because retraining is possible on the training data.

Table 1. Comparative analysis of methods, which use decision trees.

Method	*Advantages*	*Disadvantages*
CART	Rapid construction of the model. Simple interpretation (because of the model simplicity, you can easily draw a tree and observe all tree nodes).	It often comes to a local decision (for example, a hyperplane has been selected on the first step, which maximally divides the space on this step; but it will not lead to the optimal solution).
Random Forest	High quality of results, especially for data with large number of variables and small number of observations. Possibility to parallelize. No test sample is required.	Each tree is huge; as a result, obtained model is huge. Long period of model construction in order to achieve good results. Complex model interpretation (it is difficult to interpret hundreds or thousands of huge trees).
Stochastic Gradient Boosting	High quality of results, especially for data with large number of observations and small number of variables. Rather small model size (compared with the previous method), since each tree is limited to specified dimensions.	Test sample (or cross-validation) is required. Inability to parallelize well. Relatively weak resistance to erroneous data and retraining. Complex model interpretation (as in Random forest).

Common disadvantage of constructing traditional decision trees is requirement for certainty of the input data, which is achieved by applying the average values of the input parameters of the analyzed technology. This can lead to receiving significantly shifted point estimates of project performance indicators. It is also clear that the requirement for determinancy of input data is unjustified simplification of reality, because any technology is characterized by many uncertainties: uncertainty of input data, uncertainty of external environment, uncertainty associated with the nature, options and a model of project realization, uncertainty of requirements for technology effectiveness. Uncertainty factors determine technology risk, i.e. a danger of resources loss, revenue deficiency, or additional costs.

4 Neuro-fuzzy trees

In order to improve the accuracy of classification, the author suggests using neuro-fuzzy decision trees, which have property to adapt parameters by means of neural network training. In the direct cycle, fuzzy decision trees are based on fuzzy ID3 algorithm [5]. The feedback cycle, parameters of fuzzy decision trees are adapted based on stochastic gradient algorithm by traverse to the root nodes back from the leaves.

As initial data, we use the so-called triangular fuzzy numbers with a membership function of the following type (Figure 2).

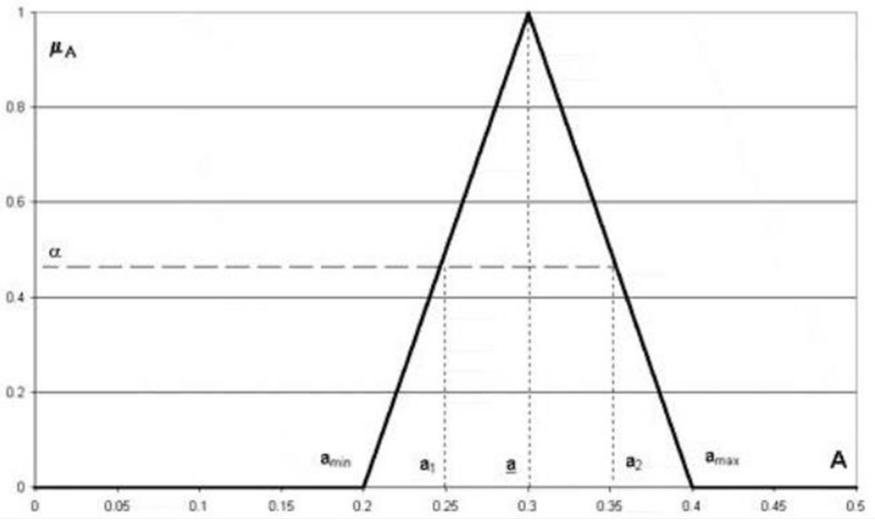

Figure 2. Membership function of a triangular fuzzy number A.

These numbers model the following statement: “Parameter **A** is approximately equal to α and is clearly in the range $[a_{min}, a_{max}]$”.

In general, a fuzzy number means a fuzzy subset of a universal set of real numbers, which have a normal and convex membership function. This description allows experts to take parameter interval $[a_{min}, a_{max}]$ as input information and most expected value of α, then the appropriate triangular number A = $(a_{min}, \alpha, a_{max})$ is built. Selection of three important points of initial data is rather common in investment analysis. Often subjective probabilities of realization of appropriate ("pessimistic", "normal" and "optimistic") initial data scenarios are compared with these points. Further, we will call parameters $(a_{min}, \alpha, a_{max})$ as *significant points* of triangular fuzzy number **A**.

We should note that attributes of technological innovative projects are classified as subjective or objective. Subjective attributes include quality features such as technical level, advantages of the enterprise, innovative risk, project management; we will estimate them by linguistic values presented by fuzzy numbers based on expert interviews.

Objective (quantitative) attributes include plans of investment costs, etc. These quantitative features are reduced to a common scale in order to provide compatibility with linguistic values of subjective features. Let us consider typical descriptions of the technology projects attributes (Table 2).

Table 2. Technology projects attributes.

№	Investment costs of the project (*10 mil. rubles), ×1	Technical level {1,2,3},×2	Advantages of the enterprise , ×3	Innovative risk, ×4	Level of the project management,×5
1.	0.82	0.0; 0.1; 0.3	average	low	low
2.	0.81	0.0; 0.1; 0.9	average	average	average
3.	0.78	0.0; 0.3; 0.5	good	high	low
4.	1.00	1.0; 0.9; 0.0	fairly poor	very low	very high
5.	0.97	1.0; 0.8; 0.6	poor	average	average
6.	0.80	0.0; 0.2; 0.9	very poor	low	average
7.	0.96	0.0;0.4; 0.9	very poor	average	very high
8.	0.78	0.0; 0.1; 0.2	very good	average	very low
9.	0.98	0.0; 0.3; 0.8	fairly good	average	average
10.	0.78	1.0; 0.7; 0.4	good	average	fairly average
11.	0.98	0.0; 0.2; 0.5	poor	low	very low
12.	0.81	0.0; 0.3; 0.9	good	average	high

The first attribute (investment costs of the project) is numerical. The third and the fifth attributes (advantages of the enterprise and level of the project management) are described in linguistic terms such as good, very poor, etc. The value of the second attribute is defined on a fuzzy set {1, 2, 3}. Innovative risk based on fuzzy classification is in linguistic terms.

Fuzzification involves conversion of attributes' numerical values to linguistic terms in order to reduce information and present it in a human-understandable form convenient for decision-making. One of the ways to determine membership functions of these linguistic variables is expert opinion or human perception. In order to automate this procedure, you can use statistical methods and fuzzy clustering based on self-organizing neural network training. Let us consider the second method.

Let there is a data set X, which must be conversed to k linguistic variables T_j, $j=1,2,...,k$. For simplicity, assume that T_j function has a form of triangulation:

$$T_1(x)=\begin{cases} 1, & x \le a_1 \\ (a_2-x)/(a_2-a_1), & a_1 < x < a_2 \\ 0, & x \ge a_2 \end{cases} \quad (1)$$

$$T_j(x)=\begin{cases}0, & x \ge a_{j+1}\\ (a_{j+1}-x)/(a_{j+1}-a_j), & a_j \le x < a_{j+1}\\ (x-a_j)/(a_j-a_{j-1}), & a_{j-1} < x < a_j\\ 0, & x \le a_{j-1}\end{cases} \quad (2)$$

$$T_k(x)=\begin{cases}1, & x \ge a_k\\ (x-a_{k-1})/(a_k-a_{k-1}), & a_{k-1} < x < a_k\\ 0, & x \le a_{k-1}\end{cases} \quad (3)$$

Parameters, which are to be defined for each attribute, form k centers $\{a_1, a_2,..., a_k\}$. A neural network algorithm – Kohonen's self-organizing maps - is an effective method for determination of these centers [3].

Let us consider a numerical attribute of the project - investment costs for a group of examples in Table 2. By Kohonen's self-organizing maps, we define: a_1= 0.68, a_2= 0.76, a_3= 0.82.

Then the membership functions for variable *x* of one of linguistic variables T_j, (j = 1, 2, 3) are described as follows:

$$T_1(x)=\begin{cases}1, & x \le 0.68\\ (0.76-x)/0.08, & 0.68 < x < 0.76\\ 0, & x \ge 0.76\end{cases} \quad (4)$$

$$T_2(x)=\begin{cases}0, & x \ge 0.82\\ (0.82-x)/0.06, & 0.76 < x < 0.82\\ (x-0.68)/0.08, & 0.68 < x < 0.76\\ 0, & x \le 0.68\end{cases} \quad (5)$$

$$T_3(x)=\begin{cases}1, & x \ge 0.82\\ (x-0.76)/0.06, & 0.76 < x < 0.82\\ 0, & x \le 0.76\end{cases} \quad (6)$$

Obviously, these linguistic terms can be described as «low», «average» and «high». The second column of Table 3 shows the degree of proximity of the attribute «investment costs» to these three membership functions.

For description of linguistic and corresponding numerical values, we assume that the membership functions of this linguistic term are known.

The measure of similarity between linguistic terms can be determined by their functions as follows:

$f(A,B) = 0.5*\{S(A,B) + S(B,A)\}$ where $S(A,B)$ and $S(B,A)$ represent degree of sub-multiplicity *A* in *B* and *B* in *A* respectively (sub-multiplicity as a degree of membership of one set to another). Here, sub-multiplicity *A* in *B* is defined as $S(A,B) = M(A \cap B)/ M(A)$, where *M* is the sum of degrees of membership of conversion of a fuzzy set to the final state.

Using the described function *f* we can calculate the degree of membership of each of two linguistic terms «Advantages of the enterprise».

The value of fuzzy attributes, for example, the attribute «technical level», can be functionally represented by a set of membership functions (Table 3). For a given set of functions, we find some new fuzzy sets, which are considered as a result of clustering of initial data for description of membership functions of set.

Table 3. Fuzzy sets of technology projects data after the neural network training.

№	Investment costs of the project (*10 mil.			Technical level {1,2,3},×2			Advantages of the enterprise , ×3			Innovative risk, ×4			Level of the project management,×5		
	low	avg.	high	low	avg.	high	good	avg.	poor	good	avg.	poor	high	avg.	low
1	0.12	0.86	0.02	0.46	1.00	0.46	0.28	0.46	0.96	0.68	1.00	0.28	1.00	0.58	0.36
2	0.00	0.92	0.08	0.56	1.00	0.56	0.38	0.97	0.42	0.36	0.56	0.92	0.54	1.00	0.58
3	0.96	0.00	0.02	0.35	0.68	1.00	0.36	0.63	0.92	0.96	0.26	0.38	1.00	0.57	0.38
4	0.00	0.00	1.00	0.92	0.78	0.36	0.85	0.13	0.08	0.88	1.00	0.32	0.18	0.22	0.76
5	0.11	0.06	0.83	1.00	0.58	0.35	0.94	0.58	0.36	0.28	0.36	1.00	0.56	1.00	0.58
6	0.12	0.31	0.57	0.88	0.42	0.24	0.36	0.95	0.48	0.98	0.38	0.27	0.56	0.38	0.56
7	0.00	0.00	1.00	0.38	0.64	0.98	0.45	0.95	0.48	0.92	0.46	0.56	0.88	1.00	0.28
8	1.00	0.00	0.00	1.00	0.57	0.32	0.26	0.42	0.98	0.26	1.00	0.38	0.57	0.38	0.59
9	0.00	0.00	1.00	0.34	0.55	1.00	0.96	0.39	0.28	0.18	0.68	0.98	0.82	0.68	0.24
10	0.08	0.86	0.06	0.56	1.00	0.56	0.44	0.29	0.95	1.00	0.45	0.56	0.26	1.00	0.56
11	0.00	0.00	1.00	0.32	0.98	0.28	0.68	0.45	0.68	0.12	0.56	0.32	0.43	0.68	0.46
12	0.46	0.54	0.00	0.96	0.56	0.25	0.98	0.27	0.36	0.78	0.56	0.68	1.00	0.36	0.68

5 Conclusion

Thus, the developed method of constructing neuro-fuzzy decision trees allows you to get rid of weighted estimations of the input data and has a property of neural network adaptation of the parameters based on stochastic gradient algorithm by traverse to the root nodes back from the leaves. In the direct cycle, fuzzy decision trees are constructed based on fuzzy ID3 algorithm. In the feedback cycle, parameters of fuzzy decision trees are adapted. Because of this strategy, the hierarchical structure of the fuzzy decision tree remains fixed.

In conclusion, we note that the proposed approach of use of back-propagation algorithm directly on the structure of fuzzy decision trees improves the accuracy of their training without damage to interpretability.

Acknowledgements

The author wishes to thank Tatiana B. Rumyantseva, National Research Tomsk State University, for assistance in the article preparation.

The paper was written as part of the research project No. 8.2.31.2015, carried out with the support of the Program “Research Foundation of Tomsk State University named after D.I. Mendeleev” in 2015 – 2016., grant RFBR No. 16-29-12858.

References

[1] S.V. Gorbachev, V.I. Syryamkin, S.A. Koynov, *Intelligent system of strategic business planning with fuzzy multiple estimation of effectiveness and risks* (LAMBERT Academic Publishing, Saarbrucken, 2012)

[2] S.V. Gorbachev, V.I. Syryamkin, I.B. Rudakov, *Recognition of complex constructed oil and gas pools based on neuro-fuzzy portraits* (LAMBERT Academic Publishing, Saarbrucken, 2013)

[3] S.V. Gorbachev, V.I. Syryamkin, *Neuro-fuzzy methods in intelligent systems of processing and analysis of multivariable information* (Publishing House of Tomsk State University, Tomsk, 2014)

[4] S.V. Gorbachev, V.I.Syryamkin, M.V. Syryamkin, *Intellectual Foresight-Forecast of Scientific and Technological Development of a State* (LAMBERT Academic Publishing, Saarbrucken, 2012)

[5] C. Z. Janikow, Proceedings of the Sixth International Symposium on AI, 360 (1993)

[6] D.V. Shashev, S.V. Shidlovskiy, V.I. Syriamkin , Yurchenko A.V. IOP Conference Series: Materials Science and Engineering **81**, 012101 (2014) doi: 10.1088/1757-899X/81/1/012101

[7] S.V. Panin, I.V. Shakirov, V.I. Syryamkin, A.A. Svetlakov, Avtometriya (1**)**, 37 (2003)