Proceedings of the 2013 Federated Conference on
Computer Science and Information Systems pp. 597–601

Image Semantic Annotation using Fuzzy Decision Trees

Andreea Popescu*, Bogdan Popescu†, Marius Brezovan‡ and Eugen Ganea§
Faculty of Automation, Computers and Electronics
University of Craiova,
Bd. Decebal 107, Craiova Romania
* Email: andreea.popescu@itsix.com
† Email: bogdan.popescu@itsix.com
‡ Email: brezovan_marius@software.ucv.ro
§ Email: ganea_eugen@software.ucv.ro

***Abstract*—One of the methods most commonly used for learning and classification is using decision trees. The greatest advantages that decision trees offer is that, unlike classical trees, they provide a support for handling uncertain data sets. The paper introduces a new algorithm for building fuzzy decision trees and also offers some comparative results, by taking into account other methods. We will present a general overview of the fuzzy decision trees and focus afterwards on the newly introduced algorithm, pointing out that it can be a very useful tool in processing fuzzy data sets by offering good comparative results.**

I. Introduction

IN TODAY'S society there is a continuous process of improvement in the wide area of knowledge acquisition, as it has a direct impact on many areas of activity. Algorithms dealing with extracting knowledge from data have as a result the decision trees and the inference procedures. The classification methods offer different results, in terms of efficiency, domains they can be applied to or ease of use.

The ID3 algorithm was initially introduced by Quinlan [14]. This algorithm offers some restrictions in terms of applicability, as it offers good results for symbolic domains, but not for numerical domains as well. [15]

Fuzzy sets have developed as an extension of the neural networks, since decisions are easier to understand when using them. They provide support for knowledge comprehensibility by offering a symbolic framework. [16] [17]

The symbolic rules together with the fuzzy logic offer complementary support for ease of understanding and modeling fine knowledge details. The fuzzy methods are today's subject in many studies, undergoing continuous improvements in order to offers good results when dealing with inexact data.

The data extraction method we propose in this paper takes into account both a fuzzy approach and a classical decision tree, being able to handle inexact data in a way that is easy to understand.

Some known studies of the fuzzy decision trees present the automatic induction of binary fuzzy trees using new discrimination quality measures. [5] The present method uses for the construction of fuzzy sets an adapted version of the ID3 algorithm. One of the methods we use as a reference is the ID3 algorithm adapted by Janikow in order to be used with fuzzy sets. [2]

II. Fuzzy Decision Trees

Decision tree structures are used to classify data by sorting it from root to leaf nodes. From the well known common tree induction algorithms we mention ID3, C4.5 or CART as they consisted reference points for our work.

In classical decision trees, nodes make a data follow down only one branch since data satisfies a branch condition, and the data finally arrives at only a leaf node.

On the other hand, fuzzy decision trees allow data to follow down simultaneously multiple branches of a node with different satisfaction degrees ranged on [0,1]. To implement these characteristics, fuzzy decision trees usually use fuzzy linguistic terms to specify branch condition of nodes. Different fuzzy decision tree construction methods have been proposed so far.[15] [14] [24]

Different papers are considering the direct fuzzy rules generation without Fuzzy Decision Tree. [26] [27] Complex techniques are used including generation of fuzzy rules from numerical data pairs, collect these fuzzy rules and the linguistic fuzzy rule base, and, finally, design a control or signal processing system based on this combined fuzzy rule base.

In [13] decision tree construction methods are incorporated into fuzzy modeling. They use the decision tree building methods to determine effective branching attributes and their splitting intervals for classification of crisp data. These intervals are then used to determine fuzzy boundaries for input variables, which will be used to form fuzzy rules. As a matter of fact, they use the decision tree construction methods for preprocessing and not for building fuzzy decision tree.

Regarding the approach in [24], the discretization of attributes is made in linguistic terms, relying on the distribution of pattern points in the feature space. Opposite to other fuzzy decision trees, this discretization to boolean form helps in reducing the computational complexity while preserving the linguistic nature of the decision in rule form. In order to minimize noise it's used pruning, resulting in a smaller decision tree with more efficient classification. The extracted rules are mapped onto a fuzzy knowledge-based network.

The rest of the paper contains the description of the proposed algorithm for fuzzy tree induction, the set of experiments and the conclusions to the current approach.

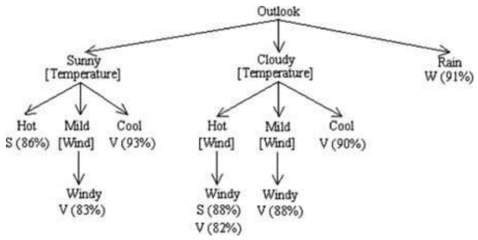

Fig. 1. Example of Generic Fuzzy Decision Tree

III. Proposed Method for Fuzzy Decision Tree Induction

A. Cluster Optimal Index Partitioning for Fuzzy Sets

A very common problem in clustering is finding the optimal set of clusters that best describe the data set. Many clustering algorithms generate a required set of clusters passed as input. In order to solve this problem, the solution would be to repetitively run the algorithm with a different set of inputs until the best schema is found.

In order to validate that, an auxiliary measure needs to be taken care of. We called this cluster optimal index.[1]

A number of cluster validity indices are described in the literature. A cluster validity index for crisp (non fuzzy) clustering is proposed by Dunn [18].

The implementation of most of these measures is very expensive computationally, especially when the number of clusters and the number of objects in the data set grow very large.

In regards to the clusters resulted by applying this mechanism, we have implemented a method of calculating the membership function of the numerical data obtained for each cluster.

The membership degree set is not a binary element from 0, 1 (as for classical decision trees), but is included in the interval [0, 1]. For each node, an attribute has a different membership degree to the current set, and this degree is calculated from the conjunctive combination of the membership degrees of the object to the fuzzy sets along the path to the node and its membership degrees to the classes, where different t-norms operators can be used for this combination.

The fuzzy decision tree induction has two major components: the procedure for fuzzy decision tree building and the generation of the fuzzy set of rules. The proposed fuzzy decision tree building procedure constructs decision tree by recursive partitioning of data set according to the values of selected attribute.

The following steps need to be implemented: attribute value space partitioning methods, branching attribute selection branching test method to determine with what degree data follows down branches of a node, and leaf node labeling methods to determine classes for which leaf nodes stand.

B. Algorithm notations and abbreviations

For better understanding of the described methodology, we have used specific notations as listed below:

- $L = \{L_1, \ldots, L_m\}$, represents the set of m classes of objects,
- $A = \{A_1, \ldots, A_n\}$, represents the set of n attributes we are taking into consideration for our analysis For each attribute we consider the following:
  - $dom(A_i)$ is the domain for the attribute A_i
  - $u^i \in dom(A_i)$, is a crisp value of attribute A_i
  - $FS_i = \{a^i_{p_1}, a^i_{p_2}, \ldots, a^i_{p_{i_k}}\}$, denotes the set of fuzzy numbers resulted after the fuzzy clustering of attribute A_i
  - we denoted FS the set of all fuzzy numbers for all attributes:
  $$FS = \{FS_1, \ldots, FS_n\}.$$
- $T = \{t_1, t_2, \ldots, t_s\}$, represent the s training objects. Each element has the following format:
  $$t_k = (u^1_k, \ldots, u^n_k, y^1_k, \ldots, y^m_k),$$
  where:
  - $u^i_k \in dom(A_i)$ is the crisp value of attribute A_i from the training object t_k
  - a single value from y^i_k is 1, the rest of them are 0 (having 1 on the i^th position means that object t_k belongs to class L_i)
- Membership degree of value $u^i_k \in t_k$ to fuzzy number $a^l_j \in FS_l$ is denoted by $\mu_{a^l_j}(u^i_k)$. For simplicity, this matching operation is denoted by T_0 operator.
  $$T_0(u^i_k, a^l_j) = \mu_{a^l_j}(u^i_k) \tag{1}$$
- $\chi = (\chi_1, \ldots, \chi_s)$ are the confidence factors of the objects from the training set ($\chi_i \in [0,1]$ represents the membership degree of object t_i from the training set T). Usually $\chi_i = 1$, $\forall i \in \{1, \ldots, s\}$.
- The Fuzzy set of the set of training objects in node N is denoted by
  $$\chi^N = (\chi^N_1, \ldots, \chi^N_s), \tag{2}$$
  where χ^N_i is the membership function of object t_i in node N.
- $I(\chi^N)$ represents the entropy of class distribution to set χ^N, in node N
- $I(\widehat{\chi}^N)$ represents the entropy of class distribution after the current node is split by attribute A_i
- $\chi^{N|a^i_j} = (\chi_1^{N|a^i_j}, \ldots, \chi_s^{N|a^i_j})$, denotes the membership degree of training objects from T to fuzzy numbers of attribute A_i, $(\chi_k^{N|a^i_j}$ represents the membership degree of

object t_k to fuzzy number $a^i_j \in FS_i$).
$\chi_k^{N|a^i_j}$ is calculated as follows:

$$\chi_k^{N|a^i_j} = T(T_0(u^i_k, a^i_j), \chi^N_k),$$

where:
- T_0 is defined in 1,
- T is a $T - norm$ operator that can be defined as follows: $T(a,b) = \min(a,b)$

- $Z_k^{N|a^i_j}$, represents the counter for examples in T belonging to class C_k and fuzzy number a^i_j of attribute A_i ($a^i_j \in D_i$). $Z_k^{N|a^i_j}$ is calculated as follows:

$$Z_k^{N|a^i_j} = \sum_{l=1}^{s} T_1(\chi_l^{N|a^i_j}, y^k_l),$$

where T_1 is a $T - norm$ operator that can be used as follows:$T_1(a,b) = a \times b$.

C. Decision Tree Node Structure

We considered a custom node structure for the extended fuzzy decision tree.

Each node N from the fuzzy decision tree is described as follows:

- F is the set of restrictions on the path from N to the root node
- V is the set of splitting attributes on the path from N to the root node
- S is the set of children of node N, when splitting is done according to attribute A_i
- χ contains the membership degree to node N

D. Fuzzy Decision Tree Induction Algorithm

In what follows it's presented a recursive algorithm for fuzzy decision tree induction of the training objects associated to the dataset we used. It is supposed that the partitioning (or clustering) mechanism of the considered attribute data is already implemented and now further used.

As described above, the numeric partitioning is done using a modified version of C-Means algorithm with additional clustering logic. The algorithm is recursive, it returns the root node and it is called for each splitting phase. Basically, at each level, after attribute partitioning, a particular attribute is selected for further splitting and branching the tree.

As already mentioned, negative information gain can also result from the t-norm (min operator) that is used in the algorithm to compute the membership degrees of the samples in a node. A negative information gain, even if it hasn't a real meaning, can lead to a correct ranking of the candidate test attributes. Instead of that, if information gain ratio is used, a negative value for the information gain cannot produce a good result. answer.

Algorithm 1 Fuzzy decision tree induction

function FUZZYTREE(m, n, s, χ, T, FS, A)
  $N \leftarrow newNode(\chi)$
  $maxGain \leftarrow 0$
  $imax \leftarrow 0$
  for $i \leftarrow 1, n$ **do**
    $Z^N \leftarrow 0$
    for $k \leftarrow 1, m$ **do**
      $Z^N_k \leftarrow 0$
      ▷ For each attribute A_i we compute $Z_k^{N|a^i_j}$ matrix, when $k = 1, m$ and $j = 1, p_{i_k}$
      for $j \leftarrow 1, p_{i_k}$ **do**
        $Z_k^{N|a^i_j} \leftarrow 0$
        for $l \leftarrow 1, s$ **do**
          $\chi_l^{N|a^i_j} \leftarrow T_0(u^i_l, a^i_j)$
          $Z_k^{N|a^i_j} \leftarrow Z_k^{N|a^i_j} + \chi_l^{N|a^i_j} \times y^k_l$
        end for
        $Z^N_k \leftarrow Z^N_k + Z_k^{N|a^i_j}$
      end for
      $Z^N \leftarrow Z^N + Z^N_k$
      $I(\hat{\chi}^N) \leftarrow 0$
      for $k \leftarrow 1, m$ **do**
        $I(\hat{\chi}^N) \leftarrow I(\hat{\chi}^N) - \frac{Z^N_k}{Z^N} \times \log_2(\frac{Z^N_k}{Z^N})$
      end for
      $I(\chi^N|A_i) \leftarrow 0$
      for $j \leftarrow 1, p_{i_k}$ **do**
        $I(\chi^{N|a^i_j}) \leftarrow 0$
        for $k \leftarrow 1, m$ **do**
          $I(\chi^{N|a^i_j}) \leftarrow I(\chi^{N|a^i_j}) - \frac{Z_k^{N|a^i_j}}{Z^{N|a^i_j}} \times \log_2(\frac{Z_k^{N|a^i_j}}{Z^{N|a^i_j}})$
        end for
        $I(\chi^N|A_i) \leftarrow I(\chi^N|A_i) + \frac{Z^{N|a^i_j}}{Z^N} \times I(\chi^{N|a^i_j})$
      end for
      $\widehat{Gain}(\chi^N, A_i) \leftarrow I(\widehat{\chi}^N) - I(\chi^N|A_i)$
      $SplitI(\chi^N|A_i) \leftarrow 0$
      for $j \leftarrow 1, p_{i_k}$ **do**
        $SplitI(\chi^N|A_i) \leftarrow SplitI(\chi^N|A_i) - \frac{Z^{N|a^i_j}}{Z^N} \times \log_2(\frac{Z^{N|a^i_j}}{Z^N})$
      end for
      $Gain(\chi^N, A_i) \leftarrow \frac{\widehat{Gain}(\chi^N, A_i)}{SplitI(\chi^N|A_i)}$
    end for
    if $Gain(\chi^N, A_i) > maxGain$ **then**
      $maxGain \leftarrow Gain(\chi^N, A_i)$
      $imax \leftarrow i$
    end if
  end for

Algorithm 2 Part 2

▷ We split node N according to attribute A_{imax}
    for $j \leftarrow 1, p_{imax}$ **do**
        for $i \leftarrow 1, s$ **do**
            $\overline{\chi}_i \leftarrow T(T_0(u_i^{imax}, a_j^{imax}), \chi_i^N)$
        end for
        $N.S_j \leftarrow$ FUZZYTREE(m, n, s, $\overline{\chi}$, T, FS, $A - \{A_{imax}\}$)
        $N.S_j.F \leftarrow N.F \cup \{[A_{imax}\ is\ a_j^{imax}]\}$
        $N.S_j.V \leftarrow N.V \cup \{A_{imax}\}$
    end for
    return N
end function

IV. Experiments

In order to demonstrate the applicability of the proposed framework we executed a wide set of experiments and verified the accuracy of the results. We have performed comparative results between the algorithm we developed, denoted here as *BFD* and two other well known similar approaches.

The other references we used were *C*4.5 [23], a well known decision tree learner based on neural networks and *NEFCLASS*, a fuzzy rule based classifier which combines fuzzy systems with neural networks. We analyzed precision and complexity for each of the 3 implementations.

For our tests we used four data sets from *UC Irvine Machine Learning Repository*[20]. You can see in Table I information related to the attributes used in the dataset we considered.

TABLE I
TEST DATASETS

data	size	attributes	classes	missing value
iris	150	4	3	no
glass	214	10 (incl. id)	7	no
thyroid	215	5	3	no
pima	768	8	2	no

The basic approach for testing was 10-fold cross validation. Data was broken into 10 sets of size n/10. We trained on 9 datasets and tested on 1. Performed this operation 10 times and considered the mean accuracy. For the algorithm we developed (*BFD*), we considered a threshold of 10% for the clustering mechanism, and since using Fuzzy C-Means, we also considered a maximum of 10 number of clusters as parameter.

In the Table II we present the average error rate $\bar{\varepsilon}$ after testing with each of the implementation.

TABLE II
ERROR RATE COMPARISON

model	iris	glass	thyroid	pima
BFD	5%	33%	5%	23%
C4.5	4%	30%	7%	31%
NEFCLASS	4%	35%	6%	29%

The precision analysis of the models considered, as seen in the table above, is good for the implementation we made, and certifies that the approach we had has good results and will offer similar results on other data sets, as part of our future work.

In terms of implementation, the algorithm was developed in *C#.NET*, and is part of a complex framework we continuously improve. The implementation decision was taken given the advantages and support that Microsoft offers for their products and the large community supporting, for best practices, performance and efficient problem solving.

V. Conclusion

This paper is aimed to introduce a new fuzzy method for handling inexact data. The approach is an extension of the classical decision trees by using fuzzy methods.

The comparative analysis we presented in this paper demonstrate that the considered approach is very solid and returned consistent results.

As observed in the above table, the precision of the proposed method is good enough and we can use it as a good reference and further integrate it in a framework we build among this approach.

The decision of considering algorithm implementation using fuzzy sets reported higher evaluation scores when focusing the training and tests on specific operational fields. We presented a novel and enhanced mechanism of image semantic annotation for segmented color images.

References

[1] B. Popescu, A. Popescu, M. Brezovan, and E. Ganea, *Unsupervised Partitioning of Numerical Attributes Using Fuzzy Sets*, Computer Science and Information Systems (FedCSIS), 2012, pp.751-754.

[2] C. Z. Janikow, *Fuzzy Decision Trees: Issues and Methods*, IEEE Trans. on Systems, Man, and Cybernetics - Part B, Vo1.28, No.1, pp.1-14, 1998.

[3] J. Jang, *Structure determination in fuzzy modeling: A fuzzy CART approach*, Proceedings IEEE Conf on Fuzzy Systems, 1994.

[4] Y. Yuan, M. J. Shaw, *Induction of fuzzy decision trees*, Fuzzy Sets and Systems, Vol.69, pp.125-139, 1995.

[5] D. Burdescu, M. Brezovan, E. Ganea, and L. Stanescu, *A New Method for Segmentation of Images Represented in a HSV Color Space*, Advances Concepts for Intelligent Vision Systems - Lecture Notes in Computer Science, Vol.5807, pp.606-617, 2009.

[6] R. Weber, *Fuzzy-ID3: a class of methods for automatic knowledge acquisition*, Proc. of 2nd Int. Conf. on Fuzzy Logic and Neural Networks(lizuka), 1992, pp.265-268.

[7] A. Gyenesei, *Fuzzy Partitioning of Quantitative Attribute Domains by a Cluster Goodness Index*, TUCS Technical Reports, 2000.

[8] J. Zeidler and M. Schlosser, *Continuous-valued attributes in fuzzy decision trees*, Proc. of the 6-th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (Granada, Spain), pp. 395-400, 1996.

[9] R. Agrawal, T. Imielinski, and A. Swami, *Mining association rules between sets of items in large databases*, Proceedings of ACM SIGMOD, 1993.

[10] R. Agrawal and R. Srikant, *Fast algorithms for mining association rules in large databases*, Proceedings of the 20th VLDB Conference, 1994.

[11] B. Popescu, A. Iancu, D. Burdescu, M. Brezovan, E. Ganea, *Evaluation of Image Segmentation Algorithms from the Perspective of Salient Region Detection* Advances Concepts for Intelligent Vision Systems - Lecture Notes in Computer Science, Vol. 6915, 2011, pp 183-194.

[12] P.K. Das and R. Bogohain, *And application of fuzzy soft set in medical diagnosis using fuzzy arithmetic operations on fuzzy number* SIBCOLTEJO, Vol. 05, pp. 107-116, 2010.

[13] T. Tani and M. Sakoda, *Fuzzy modeling by ID3 algorithm and its application to prediction of heater outlet temperature* Proc. of IEEE Int. Conf. on Fuzzy Systems (San Diego), pp.923-930, 1992.
[14] J.R. Quinlan, *Induction on Decision Trees* Machine Learning, Vol. 1, 1986, pp. 81-106.
[15] T.G. Dietterich, H. Hild and G. Bakiri, *A Comparative Study of ID3 and Backpropagation for English Text-to-Speach Mapping* Proceedings of the Interantional Conference on Machine Learning, 1990.
[16] L.A. Zadeh, *A Theory of Approximate Reasoning*. In Hayes, Michie and Mikulich (eds) Machine Intelligence 9 (1979), pp. 149-194.
[17] L.A. Zadeh, *The Role of Fuzzy Logic in the Management of Uncertainity in Expert Systems*. Fuzzy Sets and Systems, 11, 1983, pp. 199-227.
[18] J.C. Dunn, *Well separated clusters and optimal fuzzy partitions*. J. Cybern. Vol. 4, 1974.
[19] Xiaomeng Wang and Christian Borgelt, *Information Measures in Fuzzy Decision Trees*. Fuzzy Systems, 2004. Proceedings. 2004 IEEE International Conference on (Volume:1)
[20] C.L. Blake and C.J. Merz. *UCI Repository of machine learning databases*. http://archive.ics.uci.edu/ml/
[21] C. Borgelt and R. Kruse, *Graphical Models Methods for Data Analysis and Mining*. J.Wiley and Sons, Chichester, England 2002
[22] D. Nauck and U. Nauck, *Nefclass neuro-fuzzy classification (computer software)*. http://fuzzy.cs.uni-magdeburg.de/nefclass/
[23] J.R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufman, San Mateo, CA, USA 1993.
[24] S. Mitra, K.M. Konwar and S.K. Pal, *Fuzzy decision tree, linguistic rules and fuzzy knowledge-based network: generation and evaluation*. IEEE Transactions on Systems, Man, and Cybernetics, Part C, Vol. 32(4), pp. 328-339
[25] Dong-Mei Huang, *An algorithm for generating fuzzy decision tree with trapezoid fuzzy number-value attributes*. International Conference on Wavelet Analysis and Pattern Recognition, 2008, vol. 1, pp. 41-45
[26] L.-X. Wang and J.M. Mendel *Generating fuzzy rules by learning from examples*. IEEE Transactions on Systems, Man and Cybernetics, vol. 22(6), pp. 1414-1427
[27] F. Herrera, M. Lozano and J.L. Verdegay, *Generating Fuzzy Rules from Examples Using Genetic Algorithms*. Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on, pp. 340-343
[28] Tzung-Pei Hong, Kuei-Ying Lin and Shyue-Liang Wang, *Fuzzy data mining for interesting generalized association rules*. Fuzzy Sets and Systems, vol. 138(2), pp. 255-269