A New Algorithm for Optimization of Fuzzy Decision Tree in Data Mining

Abolfazl Kazemi[a,*], Elahe Mehrzadegan[b]

[a]*Assistant Professor, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*

[b]*MSc. Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*

Received 14 April, 2010; Revised 19 May, 2010; Accepted 7 july, 2010

Abstract

Decision-tree algorithms provide one of the most popular methodologies for symbolic knowledge acquisition. The resulting knowledge, a symbolic decision tree along with a simple inference mechanism, has been praised for comprehensibility. The most comprehensible decision trees have been designed for perfect symbolic data. Classical crisp decision trees (DT) are widely applied to classification tasks. Nevertheless, there are still a lot of problems especially when dealing with numerical (continuous valued) attributes. Some of those problems can be solved using fuzzy decision trees (FDT). Over the years, additional methodologies have been investigated and proposed to deal with continuous or multi-valued data, and with missing or noisy features. Recently, with the growing popularity of fuzzy representation, a few researchers independently have proposed to utilize fuzzy representation in decision trees to deal with similar situations. Fuzzy representation bridges the gap between symbolic and non symbolic data by linking qualitative linguistic terms with quantitative data. In this paper, a new method of fuzzy decision trees is presented. This method proposed a new method for handling continuous valued attributes with user defined membership. The results of crisp and fuzzy decision trees are compared at the end.

Keywords: Data mining, Classification, Decision tree, ID3, Fuzzy

1. Introduction

Data mining is known as the core stage of Knowledge Discovery in Databases (KDD), which is defined by Fayyad et al. [9] as: ‘‘the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data’’. In recent years, there is an ongoing demand for systems capable of mining massive and continuous streams of real-world data. Such systems can be used in the fields of temperature monitoring, precision agriculture, urban traffic control, stock market analysis, network security, etc. The complex nature of real-world data has increased the difficulties and the challenges of data mining in Terms of data processing, data storage, and model storage requirements [16].

Such a system can handle noise, uncertainty, and asynchrony of the real-world data [7]. Batch classification algorithms like CART [5], ID3 [26], C4.5 [28], and IFN [20] are not suitable for mining continuous data.

In almost every real-life field, one is confronted with growing amounts of data coming from measurements, simulations or, simply, from manual data registration and centralization procedures, and, most often, it would be a waste not to take advantage of these data. Recent developments in data storage devices, database management systems, computer technologies, and automatic learning techniques make data analysis tasks easier and more efficient.

In this paper, a new method of fuzzy decision trees is proposed. It is a new method for handling continuous valued attributes with user defined membership.

2. Literature Review

Many decision-tree algorithms have been developed. The most famous algorithm is ID3 that is a simple decision tree learning algorithm developed by Ross Quinlan [25] whose choice of split attribute is based on information entropy. The basic idea of ID3 algorithm is to construct the decision tree by employing a top-down, greedy search through the given sets to test each attribute at every tree node.

C4.5 is an extension of ID3 developed by Prather et al. in 1997 [24]. It improves computing efficiency, deals with continuous values, handles attributes with missing values, avoids over fitting, and performs other functions. To deal with continuous data, CART (classification and regression tree) algorithms have been proposed. CART is a data-exploration and prediction algorithm similar to

[*] Corresponding Author Email: abkaazemi@qiau.ac.ir

C4.5, which is a tree construction algorithm developed by Martinez and Suarez, 2004 [21]. Breiman et al. in 1984 summarized the classification and regression tree [5]. Instead of information entropy, it introduces measures of node impurity. It is used in a variety of different problems, such as the detection of chlorine from the data contained in a mass spectrum with Berson and Smith in 1997. CHAID (Chi-square automatic interaction detector) is similar to CART, but it differs in choosing a split node. It depends on a Chi-square test used in contingency tables to determine which categorical predictor is farthest from independence with the prediction values that proposed with Bittencourt and Clarke in 2003 [4]. It also has an extended version, Exhausted-CHAID.

In recent years, more data mining applications have been considered, some of which are mentioned in this review. For example, in 2008, Xing ping Wen Guangdao Hu Xiaofeng Yang proposed a combination of CART (classification and regression trees) and C5.0 decision tree algorithms were used to CBERS-02 remote sensing data [29]. Similarly, in another field, the investigation of possible application of decision tree in landslide susceptibility assessment was done by Nefeslioglu and Sezer and et al, in 2010 [22].

In recent years, an alternative representation has grown in popularity. It is a framework which consists of a novel fuzzy inference algorithm to generate fuzzy decision trees from induced crisp decision trees proposed by Zuhair Banda in 2006 [2]. That genetic algorithm is used to optimize and automatically determine the set of fuzzy regions for all branches and simultaneously the degree in which the inference parameters will be applied.

This representation, based on fuzzy sets and used in approximate reasoning, is especially applicable to bridging the conceptual gap between subjective/ambiguous features and quantitative data. Because of the gracefulness of gradual fuzzy sets and approximate reasoning methods used, fuzzy representation is also adequate for dealing with inexact and noisy data. Fuzzy rules, based on fuzzy sets, utilize those qualities of fuzzy representation in a comprehensible structure of rule bases.

In this paper, a new method based on partitioning the continuous-valued attributes is proposed. The proposed method could be used in most algorithms for building Decision Tree without destroying their original properties.

3. Decision Trees Algorithm

In decision-tree algorithms, examples, described by features of some descriptive language and with known decision assignments, guide the tree-building process. Each branch of the tree is labeled with a condition. To reach a leaf, all conditions on its path must be satisfied. A decision-making inference procedure (class assignments in this case) matches features of new data with those conditions, and classifies the data based on the classification of the training data found in the satisfied leaf. Tree-building is based on recursive partitioning, and for computational efficiency it usually assumes independence of all attributes. ID3 and CART are the two most popular such algorithms. While ID3 aims at knowledge comprehensibility and is based on symbolic domains, CART is naturally designed to deal with continuous domains but lacks the same level of comprehensibility.

The recursive partitioning routine selects one attribute at a time, usually the one which maximizes some information measure for the training examples satisfying the conditions leading to the node. This attribute is used to split the node, using domain values of the attribute to form additional conditions leading to sub trees. Then, the same procedure is recursively repeated for each child node, with each node using the subset of the training examples satisfying the additional condition. A node is further split unless all attributes are exhausted, when all examples at the node have the same classification, or when some other criteria are met.

The recursive tree-building can be described as follows:

1. $I_N = -\sum_K^{|C|} P_K . \log P_K$ Where C is the set of decisions and p_k is the probability (estimated from data) that an example found present in the node has classification K.
2. For each remaining attribute a_i (previously unused on the path to N), compute the information gain based.

   on this attribute splitting node N . The gain $G_i = I_N - \sum_j^{|D_i|} w_j . I_{N_j}$, where D_i denotes the set of features associated with a_i, L_N ,is the information content at the j^{th}child of N , and W_j is the proportion of N 's examples that satisfy the condition leading to that node.
3. Expand the node using the attribute which maximizes the gain.

The above tree-building procedure in fact creates a partition of the description space, with guiding principles such as having "large blocks" and unique classifications of training data in each of the blocks. It is quite natural to make classification decisions based on those partitions in such a way that a new data element is classified the same way as the training data from the same partition block. Of course, problems arise if a portion block contains training data without unique classifications. This may result from a number of factors, such as an insufficient set of features, noise or errors. Another potential problem arises if a block has no training data. This may result from an insufficient data set. Following the above intuitive decision procedure, in the inference stage a new sample's features are compared against the conditions present of the tree. This, of course, corresponds to deciding on the partition block that the new example falls into. The classification of the examples of that leaf whose conditions are satisfied by the data is returned as the algorithm's decision. For example, assuming that the shaded node contains samples with a unique decision, any other sample described in particular by the same two features "young-age" and "blond-hair" would be assigned the same unique classification.

ID3 assumes symbolic features, and any attempt to avoid this assumption trades its comprehensibility. Quinlan has extensively investigated ID3 extensions to deal with missing features, inconsistency (when a leaf contains examples of different classes), and incompleteness (when a branch for a given feature is missing out of a node).

Quinlan suggests that in tree-building, when an attribute has its information contents computed in order to determine its utility for splitting a node, each example whose needed feature is missing be partially matched, to the same normalized degree, to all conditions of the attribute.

ID3 algorithm [5] applies to a set of data and generates a decision tree for classifying the data. Fuzzy ID3 algorithm is extended to apply to a fuzzy set of data (several data with membership grades) and generates a fuzzy decision tree using fuzzy sets defined by a user for all attributes. A fuzzy decision tree consists of nodes for testing attributes, edges for branching by test values of fuzzy sets defined by a user and leaves for deciding class names with certainties. An example of fuzzy decision trees is shown in Fig. 1 below.

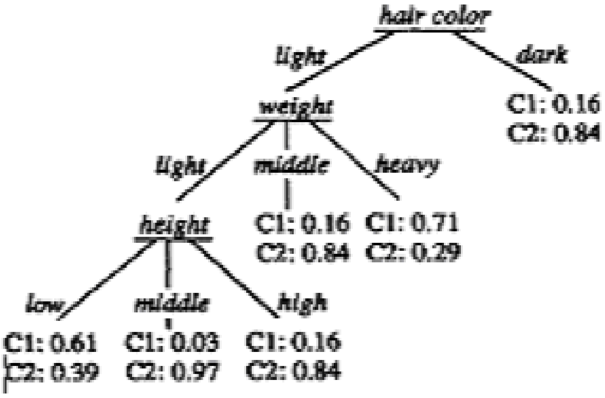

Fig. 1. Fuzzy decision tree

There are roughly a dozen publications on fuzzy decision trees (FDT). While Janikow investigates another problem of traditional DT (FDT and missing attribute values [15]), our aim is to exploit the fuzzy ID3 algorithm [27] for the addressed purposes.

4. The Proposed Algorithm

Our algorithm is very similar to ID3. However, whereas ID3 selects the test attribute based on the information gain which is computed by the probability of ordinary data, our algorithm does it by the probability of membership values for data.

The recursive tree-building can be described as defined in section 3. In the FDT described in this section, the membership function for attribute values is user defined.

We use the algorithm, which is summarized in the following:

1. Generates the root node
2. Tests for leaf node (see section 2 for three conditions)
3. Finds a test attribute

a. Divides the data according to this attribute

b. Generates new nodes for fuzzy subsets

4. Makes recursion of the process for the new nodes from point 2.

In our method, only point 3a is modified as follows:

At first, we must define the cut points. In order to choose the cut points, first, the attribute values are arranged in an ascending order. Then, we have some possible cut points between data with different classes. The flowchart of the proposed algorithm is shown in Fig. 2.

We use the following abbreviations: the information of data (D), the class information entropy E (attribute, D) after discretization with attribute, and the information gain G (attribute, D).

For attributes A {(i = 1, 2... /), find the best cut point, calculate the information gains G (Ai, D), and select the test attribute A_{max} that maximize them.

The new method will be illustrated with data from Table 1 [29]. The data D with μ are given in Table 1 too.

First the attribute values height and weight are arranged in ascending order (in table 2 and 3).

From Table 3 we select the discrimination with maximum information gain among all the candidate cut points. We get the attribute weight as the testing attribute in the root node.

Two out of three new subsets completely belong to the same class (D2 = {8} and D3 = {2, 3, 5})(table 4) . So, two leaf nodes are produced.

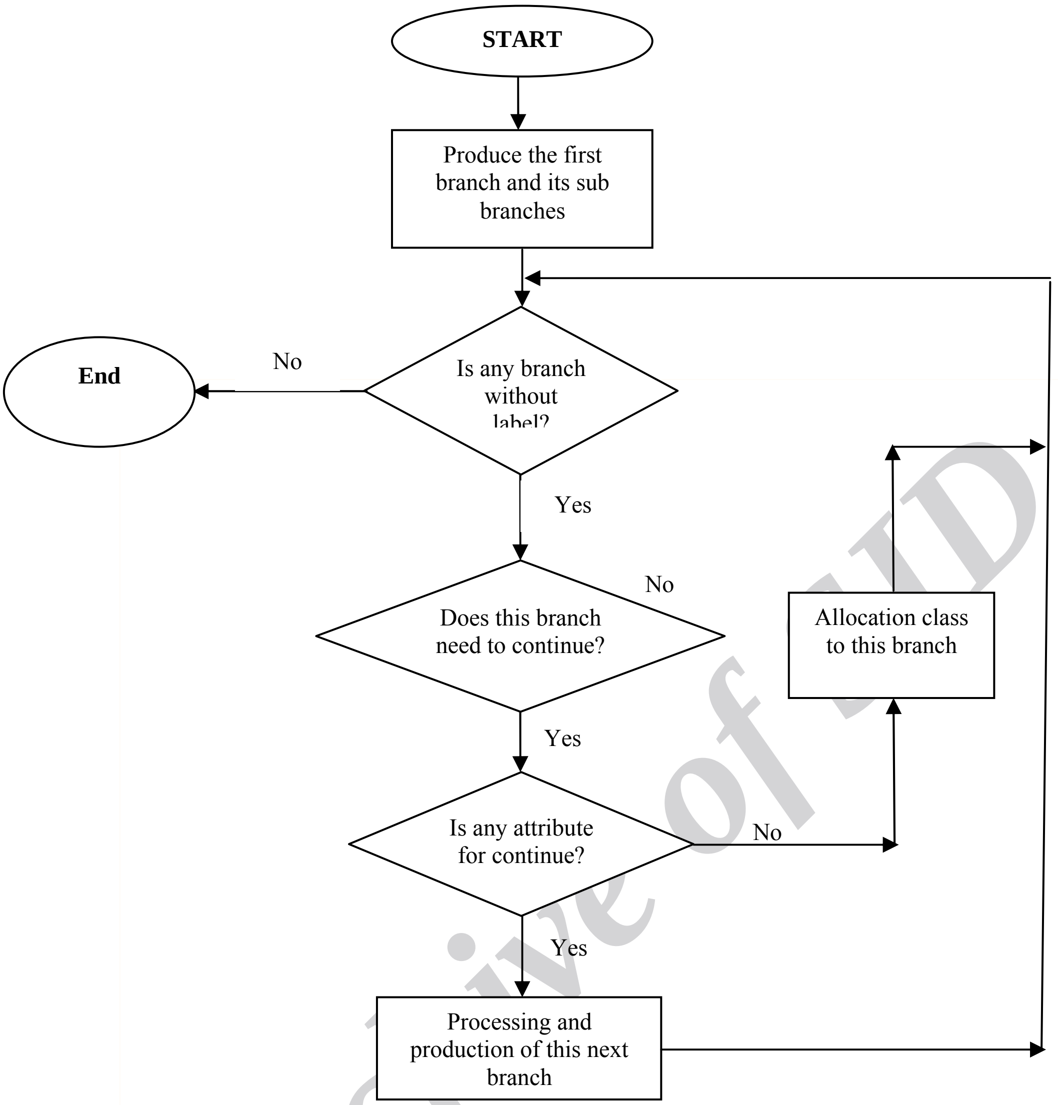
START
Produce the first branch and its sub branches
End
No
Is any branch without label?
Yes
No
Does this branch need to continue?
Allocation class to this branch
Yes
Is any attribute for continue?
No
Yes
Processing and production of this next branch

Fig 2. The flowchart of the proposed algorithm

Table 1
Data (D)

number	1	2	3	4	5	6	7	8
height	160	180	170	175	160	175	165	180
weight	60	80	75	60	75	60	60	70
hair color	blond	black	black	red	black	red	blond	blond
class	c_1	c_2	c_2	c_1	c_2	c_2	c_2	c_1
μ	1	0.8	0.2	0.7	1	0.3	1	0.5

Table 2
Data sorted by height

number	1	5	7	3	4	6	8	2
height	160	160	165	170	175	175	180	180
class	c1	c2	c2	c2	c1	c2	c1	c2
μ	1	1	1	0.2	0.7	0.3	0.5	0.8

Table 3
Data sorted by weight

number	1	4	6	7	8	5	3	2
weight	60	60	60	60	70	75	75	80
class	c_1	c_1	c_2	c_2	c_1	c_2	C_2	C_2
μ	1	0.7	0.3	1	0.5	1	0.2	0.8

Now we obtain for the *hair color* attribute:
I (D) =0.971

E(hair color , D) =

$$\frac{2.5}{5.5}\times(-\frac{1.5}{2.5}\log_2\frac{1.5}{2.5}-\frac{1.0}{2.5}\log_2\frac{1.0}{2.5})+$$
$$\frac{2.0}{5.5}\times(-\frac{0.0}{2.0}\log_2\frac{0.0}{2.0}-\frac{2.0}{2.0}\log_2\frac{2.0}{2.0})+$$
$$\frac{1.0}{5.5}\times(-\frac{0.7}{1.0}\log_2\frac{0.7}{1.0}-\frac{0.3}{1.0}\log_2\frac{0.3}{1.0})$$
$$=0.602$$

G (hair color, D) =

I (D) – E (hair color, D) = 0.369

E (height, D, cut_point_1) = 0.939

E (height, D, cut_point_2) = 0.971

E (height, D, cut_point_1_2) = 0.909

E (weight, D, cut_point_1) = 0.867

E (weight, D, cut_point_2) = 0.606

E (weight, D, cut_point_1_2) = 0.538

G (height, D, cut_point_1) = 0.032

G (height, D, cut_point_2) = 0.000

G (height, D, cut_point_1_2)= 0.062

G (weight, D, cut_point_1) = 0.104

G (weight, D, cut_point_2) = 0.365

G (weight, D, cut_point_1_2) = 0.433

Table 4
Data sorted by w

number	1	4	6	7	8	5	3	2
weight	60	60	60	60	70	75	75	80
class	c_1	c_1	c_2	c_2	c_1	c_2	C_2	C_2
μ	1	0.7	0.3	1	0.5	1	0.2	0.8

For the third subset D1 = {1, 4, 6, 7} in Table 5 we have to repeat the induction process.

Table 5
Data

number	1	4	6	7
height	160	175	175	165
weight	60	60	60	60
hair color	blond	red	red	blond
class	c_1	c_1	c_2	c_2
μ	1	0.7	0.3	1

Possible discretizations of the height and hair color attributes are shown in Table 6:

Table 6
Data sorted by *height* and *hair color*

number	1	7	4	6
height	160	165	175	175
class	c_1	c_2	c_1	c_2
μ	1	1	0.7	0.3
number	1	7	4	6
hair color	blond	blond	red	red
class	c_1	c_2	c_1	c_2
μ	1	1	0.7	0.3

I (D1) =0.987

E (height, D1, cut_point_1) = 0.623

E (height, D1, cut_point_2) = 0.960

E (height, D1, cut_point_1_2) = 0.294

E (hair color, D1) = 0.960

G (height, D1, cut_point_1) = 0.362

G (height, D1, cut_point_2) = 0.027

G (height, D1, cut_point_1_2) = 0.693

G (hair color, D1) = 0.027

The maximum information gain favors the height attribute (with two cut points) to be used for discretization in this node (in table 7).

Table 7
Data sorted by height and hair color

number	1	7	4	6
height	160	165	175	175
class	c_1	c_2	c_1	c_2
μ	1	1	0.7	0.3

We obtain two subsets with data belonging to the same class. This class gets the membership value 1.
The third subset D1,3 = {4,6} includes two inconsistent data. (Attribute values are equal but classes are different.) This inconsistence is handled by FDT. In our example, the sum of the μ values of the data in this class is equal to 1. So, the leaf node is labeled with each class and the corresponding value.
The results of the proposed algorithm using data* from the [22] compared with ID3 in the 8:
This algorithm differs from the traditional ID3 algorithm in the following ways.

- There is a membership grade j, (0 < i < 1) given for all input examples.
- The algorithm not only creates a leaf node if all data belong to the same class but also in the following cases:
    - If the proportion of a data set of a class C_K is greater than or equal to a given threshold (pre pruning of subsequent nodes because "nearly all" data belong to the same class),
    - If the number of elements in a data set is less than a given threshold (pre pruning because of “numerical tininess” of the set) or

Table 8
Results of the proposed algorithm

Data Set	#Attribute	#Classes	Samples		ID3	New-Alg
Out Look	4	3	16	# of Nodes	8	5
				Time	1.3	1
*Postoperative Patient	8	3	90	# of Nodes	30	24
				Time	4.1	3

- If there are no more attributes for classification (in ID3 there is a null class for this leaf node).
- More than one class name may be assigned to one leaf node (the real advantage of FDT).
- The fuzzy sets of all attributes are user defined. Each attribute is considered as a linguistic variable. (In our opinion, this is not necessary. The membership function can be calculated from the boundary points of the interval using the algorithm in section 4.)

5. Conclusion and Future Work

This paper proposes a method that is based on partitioning the continuous-valued attributes. The suggested method could be used in most algorithms for building DT (e.g. Fuzzy ID3) without destroying their original properties. Future work should be done in fuzzy classification of larger data sets and in investigations of more fuzzy operators for "AND" and "OR", respectively. We plan to extend our experiments with a novel cut-point-strategy. In this case, the resulting cut points depend on the density of attribute values. This algorithm can also be developed with changing the final condition for accepting the assumptions that are not given training pair.

6. References

[1] C. C. Aggarwal, J. Han, J. Wang, P.S. Yu, On demand classification of data streams. In Proceedings of KDD, 503-508, 2004.

[2] Z. Bandar, K. Crockett, D. McLean, J. O'Shea, On constructing a fuzzy inference framework using crisp decision trees. Fuzzy Sets and Systems, 157, 21, 2809-2832, 2006.

[3] P. A. Berson, Smith, S.J., Data Warehousing, Data Mining, & OLAP, McGraw Hill, New York, 1997.

[4] H. R. Bittencourt, and R.T. Clarke, “Use of Classification and Regression Trees (CART) to Classify Remotely-Sensed Digital Images”, Proceedings of International Geosciences and Remote Sensing Symposium, IGARSS ’03, Vol. 6, pp. 3751-3753, 2003.

[5] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and Regression Trees. CRC Press, Boca Raton, FL, 1984.

[6] J. Catlett, on changing continuous attributes into ordered discrete attributes. Springer, Berlin, Heidelberg, Porto, Portugal, 1991.

[7] T. M. Cover, Elements of Information Theory, second ed. Wiley- Interscience, New York, NY, 2006.

[8] D. E. Culler, W. Hong, Wireless sensor networks - introduction. Communications of the ACM 47 (6), 30–33, 2004.

[9] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth: From Data Mining to Knowledge Discovery: An Overview. Advances in knowledge discovery and data mining book contents, 1-34, 1996.

[10] U. M. Fayyad, K.B. Irani, Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, Machine Learning, 13th IJCAI, vol. 2, Chambery, France, Morgan Kaufmann, 1022-1027, 1993.

[11] U. M. Fayyad, K.B. Irani, On the handling of continuous-valued attributes in decision tree generation. Machine Learning, 8, 87-102, 1992.

[12] C. Giraud-Carrier, A Note on the Utility of Incremental Learning. AI Communications, 13, 4, 215-223, 2000.

[13] P. Helmbold, P.M. Long, Tracking drifting concepts by minimizing disagreements. Machine learning, 14, 27-46, 1994.

[14] S. Hettich, S.D. Bay, The UCI KDD archive. Department of Information and Computer Science. University of California, Irvine, CA. http://kdd.ics.uci.edu, 2006.

[15] C. Z. Janikow, Fuzzy processing in decision trees. In Proceedings of International Symposium on Artificial Intelligence, 360-367, 1993.

[16] H. Kargupta, Distributed Data Mining for Sensor Networks Tutorial, ECML/PKDD, Pisa, 2004.

[17] N. Kasabov, Adaptation and interaction in dynamical systems: Modeling and rule discovery through evolving connectionist systems. Applied Soft Computing, 6, 3, 307-322, 2006.

[18] M. Last, A. Kandel, O. Maimon, E. Eberbach, Anytime algorithm for feature selection. Lecture Notes in Computer Science, 532–539, 2000.

[19] M. Last, O. Maimon, A Compact and Accurate Model for Classification. IEEE Transactions on Knowledge and Data Engineering, 16, 2, 203-215, 2004.

[20] O. Maimon, M. Last, Knowledge Discovery and Data. Mining- The Info-Fuzzy Network (IFN) Methodology. Kluwer Academic Publishers, December 2000.

[21] G. Martinez-Munoz, A. Suarez, Aggregation ordering in bagging, in: International Conference on Artificial Intelligence and Applications (IASTED),Acta Press, 2004.

[22] H. A. Nefeslioglu, E. Sezer, C. Gökçeoğlu, A.S. Bozkır, T. Y. Duman, Assessment of landslide susceptibility by decision trees in the metropolitan area of Istanbul, Turkey. Mathematical Problems in Engineering, 2010.

[23] D. J. Newman, S. Hettich, UCI Repository of machine learning databases [http://www.ics.uci.edu/mlearn/MLRepository.html] (Irvine, CA: University of California, Department of Information and Computer Science, 1998).

[24] J.C. Prather, D.F. Lobach, L.K. Goodwin, J.W. Hales, M. L. Hage and W.E. Hammond, Medical data mining: knowledge discovery in a clinical data warehouse, in Proceedings of the 1997 AMIA Annual Fall Symposium, Nashville, Tennessee, USA, 101-105. 1997,

[25] J. R. Quinlan, Induction of Decision Trees. Machine Learning, 1, 1, 81-106, 1986.

[26] J. R. Quinlan, Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993.

[27] G. Seidelmann, Using heuristics to speed up induction on continuous-valued attributes. In P. B. Brazdil, editor, Proc. of 6 th European Conference on Machine Learning, pages 390- 395. Springer, Berlin, Heidelberg, Vienna, Austria, 1993.

[28] M. Umanol, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu, J. Kinoshita, Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems. Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on 26-29 Jun, 3, 2113-2118, 1994.

[29] W. Xingping, H. u. Guangdao, Remote Sensing Data Mining Using Decision Tree Algorithm. First International Workshop on Knowledge Discovery and Data Mining, 2008.