International Journal of Engineering & Scientific Research (IJMRA Publications)
Vol. 6 Issue 4, April 2018,
ISSN: 2347-6532 Impact Factor: 6.660
Journal Homepage: http://www.esrjournal.com Email: esrjeditor@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

A Survey: Underlying Techniques of Decision Tree

Vivek Kumar Soni*
Mr. Satish Pawar**

Article Received: 4th March, 2018
Article Revised: 15th March, 2018
Article Accepted: 25th March, 2018

Keywords:

Decision Tree;
Pruning;
Splitting;
Stopping.

Abstract

Decision tree is one of the simplest and widely used classification methods because it has least complexities but simple to use, understand and built. Therefore analysts from various disciplines prefer this algorithm. Decision tree can be viewed as an extended version of IF-THEN clause with inherited multi-levels of "IF" followed by a single "THEN". In this tree all levels of the "IF" forms rules (or patterns) and "THEN" gives the consequence. This paper presents a brief survey of popular and current techniques used in construction of decision tree, which also enlists some journey details of decision tree from historical to modern view. After considering potential advantages of DTC's over single-state classifiers, the subjects of structure design, feature selection at each internal node are discussed. To achieve higher performance cutting off the unimportant edges of decision tree, some pruning methods are also discussed. Finally we mentioned some comparative details among the splitting criterion and among pruning methods.

Author correspondence:

Vivek Kumar Soni,
Research Scholar, Department of Computer Science and Engineering
Samrat Ashok Technological Institute, Vidisha-464001 (M.P.) India
Corresponding Author Email Id: viveksoni2@gmail.com

1. Introduction

With the advancement in the communication technologies, existing and newly generated data are rolling over. These data can be utilized for various purposes to extract meaningful information. To extract desired information from these available data we need to analyze it. Because the size of this data is very large, that the manual analysis of it is almost impossible. Therefore analysis requires some kind of automation which can save human efforts and precious resources as well as gives optimized results. Machine learning [1], [2] and data mining [3], [4] are such two multi-stepped procedure which

* Doctorate Program, Linguistics Program Studies, Udayana University Denpasar, Bali-Indonesia
** STIMIK STIKOM-Bali, Renon, Depasar, Bali-Indonesia

can be used to accomplish this task. In machine learning we build a learning model which is trained based on either some predefined rules (called supervised learning) or uses properties of object at run time (the time of actual use) (called unsupervised learning). There is another form of machine learning known as reinforcement learning which is based on the behavior based psychology of software agents [5]. Reinforcement learning is very important learning used in several theories like game, control, information theory, and operations research.

In data mining, to build model we use one of the machine learning methods discussed above [6], [7]. Classification, prediction, and clustering, etc. are some data mining methods of which first two methods uses supervised leaning whereas the last one uses unsupervised learning respectively. Among the various data mining methods, classification is widely used because it is based on supervised learning, easy to implement, and performance wise it is very efficient.

There are several classification method such as decision tree, support vector machine, naive bays, rule based etc. Decision trees are like a general rooted tree which is recursive in nature. Top-down approach is used to build decision trees. Some superficial properties of decision tree are listed below:

- decision trees may have 1...n number of branches
- each internal node represents attribute
- each branch is labeled with attribute values
- external nodes are mounted with class values

Requirements for constructing decision trees model are given below:

- structured dataset
- some splitting criteria
- stopping criteria
- pruning criteria (optional)
- building and training the model
- testing the model
- performance evaluation

As the data are gathered from various resources which may be heterogeneous in nature for analysis, therefore we need some mechanism to make the homogeneity among these data. Also there is a requirement to transform the data into the structured (machine like) data so that it can be feed in to the learning machine. By using the splitting criteria we obtained the most eligible attribute to partition the dataset. Splitting criteria is also termed as attribute selection measure. With the introduction of decision tree splitting criteria plays a very important role in the construction of the decision tree. There may be a situation when the tree suffers from the problem of over-fitted, then a good pruning method is the potential solution for this. While constructing a decision tree, there must be a condition that is responsible for stopping the tree growth either normally or abnormally.

From starting till now, the decision tree has witnessed several improvements and modifications in itself [8]-[10]. Therefore in this paper, we studied the journey of decision tree in terms of attribute selection measure and splitting criteria and what enhancements and modifications have done in it.

In the forthcoming sections, we described various splitting criterion, pruning techniques, stopping criteria.

2. Splitting Criteria

Splitting criteria decides how the given dataset will be partitioned so that the pattern searching is very fast and successful. There are various several splitting criteria are available in the universe. The brief description of some of them is given below.

Univariate Splitting Criteria

When a node is split on the basis of single attribute values, then it is termed as univariate [3], [4]. In most of the methods, splitting criteria are univariate. Therefore, among all the attributes, one which reflects higher information is used to split. There are a number of univariate splitting criteria with each

have different nature such as: dependence, information theory, distance, purity based, impurity based, normalization based, and binary based. The next section describes some of these criteria.

Purity based Splitting Criteria

Suppose *D* is a dataset contains *A* number of attributes where $A = \{a_1, a_2, a_3 ..... a_n\}$

The purity based criteria is a probability based function which gives how much information dataset *D* contains [3]. For example, suppose p_i is a probability that attribute a_i contains information. Purity based function can be formulized as follows:

$$E_{purity}(D) = \sum_{i=1}^{n} p_i$$

Where $1 \geq i \leq n$, and $E_{purity}(D)$ is the function (purity function) information contained by dataset *D*.

Imputiry based Splitting Criteria

Impurity based criteria is just the reverse process of purity function [3]. Sometimes it is good to use impurity based criteria such as in information gain (IG), gini index etc. Impurity based function can be formulized as follows:

$$E_{impurity}(D) = \sum_{i=1}^{n} 1 - p_i$$

where $1 \geq i \leq n$ and $E_{impurity}(D)$ is the function (impurity function) information contained by dataset *D*.

Information Gain

J.R. Quinlan [13] proposed information gain for ID3 decision tree induction. The mathematical notation for splitting criteria is given below:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2 p_i$$

Where p_i is the probability that an instance of dataset *D* belongs to class C_i and is calculated by $|C_{i,D}|/|D|$. A *log* function to the base *2* is used, because the information is encoded in bits. *Info(D)* is the entropy of the whole dataset *D*. Now suppose an attribute *A* of *D* has *v* distinct values, therefore *D* can be partitioned in to *v* distinct subsets. The entropy of each attribute is calculated by the following formula:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

The information gain (IG) for each attribute can be calculated as follows:

$$Gain(A) = Info(D) - Info_A(D)$$

An attribute with the highest Gain is used to split the dataset i.e. used to make a node of a decision tree.

Gain Ratio

A successor of information Gain introduced by Quinlan [14], which uses information gain to find the gain ratio for attribute as follows:

$$GainRatio(A) = \frac{Gain(A)}{Info_A(D)}$$

As the value of denominator goes low, the gain ration tries to favor that attribute, and when the denominator becomes zero then the Gain Ratio cannot be defined. The attribute which scored highest gain ratio is opted. Quinlan [15] showed that the Gain Ratio outperforms information gain both from the complexity and accuracy aspects.

Gini Index

Gini Index [16], [17] calculates the impurity of a dataset *D* as follows:

$$Gini(D) = 1 - \sum_{i=1}^{n} p_i^2$$

If a split on *A* partitions *D* into $D_1, D_2, \ldots D_k$ partitions the gini index of D given that partitioning is

$$Gini_A(D) = \sum_{j=1}^{k} \frac{|D_j|}{|D|} Gini(D_j)$$

Consecutively, the attribute selection criterion is defined as follows:

$$Gini(A) = Gini(D) - Gini_A(D)$$

Chi-Squared Statistics

Chi square statistic is used to find the relationship between two categorical variables [18]. Chi square statistic output a difference between the observed counts and expected counts i.e. it compares the expected value with the values actually collected. One of the common form to calculate chi-squared statistics is:

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

Where *O* is the observed value, *E* is the expected value, and *i* is the i^{th} instance.

Normalized Impurity based Criteria

The impurity based criteria has a tendency to choose attribute with many values, therefore there is a need of some mechanism such as normalization to offer the equal opportunity to all the attributes [19].

Binary Criteria

Using binary criteria, any internal node of the decision tree will always have at most two branches. In this method the given domain is divided into two sub-domains.

Let $B^*(A_i, D, d_1, d_2)$ denote the binary criterion value for attribute A_i over dataset *D* when d_1 and d_2 are its corresponded sub-domains. The value obtained by division of attribute domain into two sub-domains, is used for comparing attributes, namely

$$B^*(A_i, D) = B(A_i, D, d_1, d_2)$$

Such that,

$$d_1 \cap d_2 = \phi$$
$$d_1 \cup d_2 = D$$

Twoing Criteria

Breiman [16] showed that the Gini index performance's lacks when domain of the target attribute is relatively wide. To overcome from this problem, they suggested to use a binary criterion know as twoing criterion. The Twoing criterion has also been built in CART (Salford systems, 1995).

$$(x, S, n) = 0.25 P_L P_R \left[\sum_{j=1}^{c} |P_{j/L} - P_{j/R}| \right]^2$$

Where $P_L \in d_1$, $P_R \in d_2$ are the probability distributions belongs to class *j*. The $P_L P_R$ is designed such that the relative splitting is even. The maximum value for this factor is *0.25* when $P_L = P_R = 0.5$, if any of these proportions are close to *0* or *1* then it declines. If the values are less than or equal to any pre-specified value *x* then it is assigned to LHS node *n* otherwise RHS node *n*. When the target attribute is binary, the twoing and Gini criteria are same.

Minimum Description Length (MDL)

MDL is an information theoretic system selection principle [20]-[22]. MDL believes that the simple and compact representation of the dataset is the best and most likely explanation of the dataset. The notation for MDL is given below:

$$\forall P, \exists C \text{ such that } Len(C(x)) = -\log_2 P(x)$$

Where *P* is the probability distribution, *C* is the code corresponding to *P* and *Len(C(x))* is the length (minimum code length) of *C(x)* in bits.

The above MDL coding equation is conversely true too.

For multivalued attributes, MDL gives least biased results in attribute selection measure. The encoding technique exploited by MDL defines "optimized" decision tree. This outputs a simplest decision tree.

Other Univatiate Criteria

Several other univariate splitting criteria are present in the literature such as orthogonality (ORT) [23] which performs better than IG and Gini index for specific problems. Kolmogorov-Smirnov [24-26] outperforms the Gain ratio criteria. CHAID (Chi-square automatic interaction detector) [18] is a decision tree algorithm based on statistical χ^2 for splitting. In certain cases performance of C-SEP is better than Gain and Gini index. G-statistic is an information theoretic criterion which is a close approximation of χ^2 distributions. Permutation statistic [21], mean posterior improvement [22], and hypergeometric distribution measure [23] are also some remarkable splitting criterions.

Comparison of Univatiate Splitting Criteria

Several researches have conducted comparative study for the splitting criteria, of them, most comparisons are based on experimental outcomes and some of them are based on theoretical conclusions [16], [23], [27]-[33]. Most of the researchers deduced that the performance of the decision tree is not much affected by the splitting criteria. Each criterion outperforms in some cases but under performs in some other cases. So the choice of splitting criteria is strictly depend upon the domain of data.

Multivatiate Splitting Criteria

In a multivariate splits, more than one attribute participates in splitting process for a node rather than a single attribute. Most of the multivariate splits are based on the linear combinations of features such as CART. In this, new attributes are explored based on the existing ones. To find the best multivariate splitting criteria is more complicated than finding the best univariate splitting criteria. The linear combinations can be obtained by using the methods such as greedy searching [16], [34], linear programming methods [35], [36], linear discriminant analysis [35], [37]-[41] and several others [42]-[44]. Although the multivariate splitting criteria may dramatically improve the performance of the decision tree but because of complication multivariate criteria is less popular than the univariate criteria.

3. Stopping Criteria

There must be some criteria that ensure the stopping of growing of a decision tree. Some common halting criteria are listed below:

- Uniclass training set.
- All attributes has been tested.
- Maximum depth of a tree has been reached.
- The minimum number of elements in the leaf node is less than the minimum number of elements in the non-leaf node

4. Tree Pruning Methods

Tight stopping criteria results small and/or under-fitted decision trees, whereas loose stopping criteria results large and/or over-fitted decision tree. Pruned trees are smaller, less complex and, thus, easy to comprehend. They are faster, better, and more accurate. Breiman et al. [16] was the original introducer of pruning methods. There are two general approaches of tree pruning: *prepruning* and *postpruning* [3].

In prepruning, growing of a tree is stopped by deciding not to further split the subset of training instances at a given node. Therefore the node becomes a leaf node. Information gain and gini index

can make use of prepruning. Postpruning method let the construction of tree complete and then remove subtree from that. At a given node a subtree is removed by removing all the branches of a node and making that a leaf node. Breiman et al. [16] tree pruning criteria is based on postpruning approach.

The following section describes some most common pruning techniques:

Weakest Link Pruning

This is a post pruning approach (also known as cost complexity pruning and error complexity pruning) [16]. It employees bottom-up technique. For each internal node *N*, for each and every subtree of *N* it computes cost complexity when tree was pruned and the cost complexity for all branches of a node *N* when tree wasn't pruned. These cost complexities are compared and the tree with higher cost complexity is discarded. In this approach the cost complexity of a tree is the function of the number of leaves in the tree and error rate of the tree.

Pessimistic Pruning

Quinlan's pessimistic pruning approach [14] makes use of error rate of a tree to decide about the tree pruning. To calculate error rate it requires training set (instead of prune set as in cost complexity method). Estimation of accuracy or error rate based on training set is overly optimistic and therefore strongly biased. The pessimistic pruning approach is therefore adjusts the accuracy or error rate calculated from the training set.

Minimum Description Length Pruning

To define the best decision tree, minimum description length (MDL) pruning method uses encoding techniques [45]. MDL states that the best decision tree is one which requires least number of bits to:

- encode the tree
- encode exception to the tree (i.e. cases that are not correctly classified by the tree).

The main idea of MDL is to give priority to the simplest solution. Instead of using the error rate to prune tree, MDL makes use of number of bits to encode the tree for that. The best pruned tree is the one that require fewer number of encoding bits.

Reduced Error Pruning

Quinlan [13] proposed that While traversing a tree from bottom to root, at each internal node apply check to decide whether it can be pruned or not. The procedure checks that whether the tree's accuracy is reduced or not by the replacement of a node by a most popular class. If the tree's accuracy is not reduced then the node is pruned. This process continues until and unless any further pruning decreases the tree's accuracy. This process results the smallest subtree with higher accuracy. The advantage of reduced error pruning is speed and simplicity.

Minimum Error Pruning

Niblett and Bratko [46] proposed minimum error pruning approach which follows the bottom up traversal technique. In bottom up traversal, at each non-leaf node it compares the 1-probability error rate of a tree with and without pruning. Where, the 1-probability error rate is the correction in the simple probability error rate by using the frequencies. A node is pruned if it does not increase the 1-probability error rate.

Error Based Pruning

Hall, Collins, Bowyer, and Banfield [47] proposed the simplest decision tree pruning method. At a node this method uses error rate derived from training set and does not require test set error rate. To control the pruning, error-based pruning (EBP) uses two parameters named binomial distribution and certainty factor (CF) which is followed by error rate. The higher value of CF indicates that the current error rate is acceptable and no pruning is required and vice versa.

Optimal Pruning

Bratko and Bohanec [48] and Almuallim [49] proposed optimal pruning algorithm (OPT). OPT finds the optimal pruned tree T_* where,

- initial decision tree T_0 whose accuracy $a(T_0) = 1$ and
- required minimal accuracy of the pruned tree $\hat{a} \in [0,1]$.

Smallest pruned tree of T_0 is T_* which satisfies the condition $a(T_*) > a$. Among the multiple solutions (if there is), it finds one.

From the fully-constructed initial decision tree, this algorithm finds the sequence (based on size) of pruned subtrees. A sequence always contains optimal pruned tree which can easily be located. OPT is based on dynamic programming and is recursive in nature.

Critical Value Pruning

John Mingers [50] proposed critical value tree pruning method. He states that attribute importance is determined by the split criteria and the chosen attribute decides how well the data will be classified at the node. In this method a critical value is specified. Nodes which do not reach the critical value are pruned. Larger the critical value implies smaller decision tree and larger pruning. The estimation of critical value depends on the criteria used in creating the tree.

Other Pruning Methods

In the literature there are several other pruning methods. Minimum message length (MML) pruning [51] generates the smallest overall pruned tree. Theoretically-justified pruning [52] method scans the tree in bottom up manner and at each node it decides whether the subtree should be kept or delete. chi-squared pruning [18] method is the prepruning algorithm which requires observed and expected results for pruning. Fast bottom up decision tree pruning [46] proposed that the subtree pruning decision is completely based on that subtree only. Decision tree pruning using backpropagation neural networks [53] states that some of the removed nodes from the overfitted decision tree may have contribute in classification of newly incoming data. Therefore, in place of absolute removal of such nodes backpropagation neural network method assigns weight to them according to their importance.

Pruning Methods Comparisons

Aim of several studies is for comparison of various pruning methods [13], [29], [54]. The experimental results showed that no pruning method outperforms the other i.e. some pruning methods tend to over-pruning and some of the pruning methods tend to under-pruning. Cost complexity and error pruning techniques produce over-pruned tree which is smaller in size but degrades the accuracy. EBP, PEP, and MEP are suffering from over-pruning.

5. Conclusion

The paper presents an overview of important ingredients (splitting criteria, stopping criteria, and pruning criteria) used in decision tree construction. Each and every algorithm has a different techniques but the aim is to produce optimal result. Most of the algorithms can be employed within a single frame but the difference is in their use of combinations.

References

[1] R. Kohavi, F. Provost, "Glossary of terms," Machine Learning, vol. 30, no. 2-3, pp. 271-274, 1998.

[2] U.V Kulkarni, S.V Shinde, "Neuro –fuzzy classifier based on the Gaussian membership function",4th ICCCNT 2013, July 2013,Tiruchengode,India.

[3] Jiawei Han, Micheline Kamber, "Data Mining Concepts and Techniques", 2nd ed, M. K. Publishers, 2006.

[4] Jiawei Han, Micheline Kamber, Jian Pei, "Data Mining: Concepts and Techniques", 3rd ed., Morgan Kaufmann Publishers, 2012.

[5] Madan Somvanshi, Pranjali Chavan, Shital Tambade, S.V. Shinde, "A Review of Machine Learning Techniques using Decision Tree and Support Vector Machine", Proc. IEEE

International Conference Computing Communication Control and automation (ICCUBEA 2016), Aug. 2016.
[6] Shahrukh Teli, Prashasti Kanikar, "A Survey on Decision Tree Based Approaches in Data Mining", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 5, Issue 4, pp 613-617, 2015.
[7] Lior Rokach and Oded Maimon, " Top Down Induction Of Decision Tree Classifier-A Survey ", IEEE Transaction On System, Man and Cybernetics Part C, Vol 1, No. 11, Nov.2002.
[8] S. R. Safavin and D. Landgrebe, "A survey of decision tree classifier methodology," IEEE Trans. Syst., Man, Cybern., vol. 21, no. 3, pp. 660–674, Jul. 1991.
[9] S. K. Murthy, "Automatic construction of decision trees from data: a multidisciplinary survey," Data Mining Knowl. Disc., vol. 2, no. 4, pp. 345–389, 1998.
[10] R. Kohavi and J. R. Quinlan, "Decision-tree discovery," in Handbook of Data Mining and Knowledge Discovery, W. Klosgen and J. M. Zytkow, Eds. London, U.K.: Oxford Univ. Press, ch. 16.1.3, pp. 267–276, 2002.
[11] Rokach, Lior; Maimon, O., "Data mining with decision trees: theory and applications". World Scientific Pub Co Inc. , 2008, ISBN 978-9812771711.
[12] Richard G.Brereton, "Consequences of sample size, variable selection, and model validation and optimisation, for predicting classification ability from analytical data", Elsevier, Vol 25, Issue 11, pp 1103-1111, Dec 2006.
[13] J. R. Quinlan, "Simplifying decision trees," Int. J. Man-Mach. Studies, vol. 27, pp. 221–234, 1987.
[14] J. R. Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers Inc. San Francisco, CA, USA Vol. 16, Issue 3, pp 235-240, 1993.
[15] J. R. Quinlan, "Decision trees and multivalued attributes," in Machine Intelligence, J. Richards, Ed. London, U.K.: Oxford Univ. Press, vol. 11, pp. 305–318, 1988.
[16] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and Regression Trees", Belmont, CA: Wadsworth, 1984.
[17] S. B. Gelfand, C. S. Ravishankar, and E. J. Delp, "An iterative growing and pruning algorithm for classification tree design," IEEE Trans. Pattern Anal. Mach. Intell., vol. 13, no. 2, pp. 163–174, Feb. 1991.
[18] F. Attneave, "Applications of Information Theory to Psychology", New York: Holt, Rinehart and Winston, 1959.
[19] J. R. Quinlan, "Induction of decision trees," Mach. Learn., vol. 1, pp. 81–106, 1986.
[20] Rissanen, J., "Modeling by shortest data description", Automatica, Vol.14 Issue 5, pp 465–658, 1978.
[21] Barron A, Rissanen J, Yu B, "The minimum description length principle in coding and modeling.", IEEE Trans Inf Theory, Vol. 44, Issue 6, Oct. 1998.
[22] Grunwald, P, "the Minimum Description Length principle". MIT Press, June 2007.
[23] U. M. Fayyad and K. B. Irani, "The attribute selection problem in decision tree generation," in Proc. 10th Nat.Conf. Artificial Intelligence, Cambridge, MA, pp. 104–110, 1992.
[24] J. H. Friedman, "A Recursive Partitioning Decision Rule for Nonparametric Classification," IEEE Trans. on Computers., vol. C26, no. 4, pp. 404–408, Apr. 1977.
[25] E. Rounds, "A combined nonparametric approach to feature selection and binary decision tree design," Pattern Recognition., vol. 12, pp. 313–317, 1980.
[26] P. E. Utgoff , J. A. Clouse, "A Kolmogorov-Smirnoff Metric for Decision Tree Induction," Dept. Comp. Sci., Univ. Massachusetts, Amherst, Tech. Rep. no. 96-3, Jan. 1996.
[27] E. Baker , A. K. Jain, "On feature ordering in practice and some finite sample effects," in Proc. 3rd Int. Joint Conf. Pattern Recognition, San Diego, CA, pp. 45–49, 1976.
[28] Ben Bassat, "Myopic policies in sequential classification," IEEE Trans. Comput., vol. C–27, no. 2, pp. 170–174, Feb. 1978.
[29] J. Mingers, "An empirical comparison of pruning methods for decision tree induction," Mach. Learn., vol. 4, no. 2, pp. 227–243, 1989.
[30] W. L. Buntine, T. Niblett, "A further comparison of splitting rules for decision-tree induction," Mach. Learn., vol. 8, pp. 75–85, 1992.

[31] T. Loh, T. Shih, "Split selection methods for classification trees," Statistica Sinica, vol. 7, pp. 815–840, 1997.
[32] S. L. Loh, S. L. Shih, "Families of splitting criteria for classification trees," Statist. Comput., vol. 9, pp. 309–315, 1999.
[33] S. L. Shih, "Selecting the best splits classification trees with categorical variables," Statist. Probability Lett., vol. 54, pp. 341–345, 2001.
[34] S. K. Murthy, S. Kasif, S. Salzberg, "A system for induction of oblique decision trees," J. Artif.Intell. Res., vol. 2, pp. 1–33, Aug. 1994.
[35] R. Duda, P. Hart, "Pattern Classification and Scene Analysis", New York: Wiley, 1973.
[36] P. Bennett, O. L. Mangasarian, "Multicategory discrimination via linear programming," Optimization Meth. Softw., vol. 3, pp. 29–39, 1994.
[37] J. H. Friedman, "Arecursive partitioning decision rule for nonparametric classifiers," IEEE Trans. Comput., vol. C26, no. 4, pp. 404–408, Apr. 1977.
[38] J. Sklansky, G. N. Wassel, "Pattern Classifiers and Trainable Machines", New York: Springer-Verlag, 1981.
[39] Y. K. Lin, K. Fu, "Automatic classification of cervical cells using a binary tree classifier," Pattern Recognition., vol. 16, no. 1, pp. 69–80, 1983.
[40] W. Y. Loh, N. Vanichsetakul, "Tree-structured classification via generalized discriminant analysis," J. Amer. Statist. Assoc., vol. 83, pp. 715–728, 1988.
[41] G. H. John, "Robust linear discriminant trees," Learning From Data: Artificial Intelligence and Statistics V. D. Fisher and H. Lenz, Eds. New York: Springer-Verlag, ch. 36, pp. 375–385, 1996.
[42] P. E. Utgoff, "Perceptron trees: a case study in hybrid concept representations," Connect. Sci., vol. 1, no. 4, pp. 377–391, 1989.
[43] D. Lubinsky, "Algorithmic speedups in growing classification trees by using an additive split criterion," in Proc. AI Statistics, pp. 435–444, 1993.
[44] I. K. Sethi, J. H. Yoo, "Design of multicategory, multifeature split decision trees using perceptron learning," Pattern Recognition., vol. 27, no. 7, pp. 939–947, 1994.
[45] J. Rissanen, "Stochastic Complexity and Statistical Inquiry Theory", Singapore: World Scientific, 1989.
[46] T. Niblett, I. Bratko, "Learning decision rules in noisy domains," Expert Systems. Cambridge, MA: Cambridge Univ. Press, 1986.
[47] L.O. Hall, R. Collins, K.W. Bowyer, R. Banfield, " Error-based pruning of decision trees grown on very large data sets can work!", in Proc. 14th Int. Conf. Tools with Artificial Intelligence, IEEE Xplore, Washington DC, USA 2003.
[48] I. Bratko, M. Bohanec, "Trading accuracy for simplicity in decision trees," Mach. Learn., vol. 15, pp. 223–250, 1994.
[49] H. Almuallim, "An efficient algorithm for optimal pruning of decision trees," Artif. Intell., vol. 83, no. 2, pp. 347–362, 1996.
[50] John Mingers, "An Empirical Comparison of Pruning Methods for Decision Tree Induction", Machine Learning, Vol. 4, Issue 2, pp 227-243, Nov. 1989.
[51] C. Wallace, J. Patrick, "Coding decision trees," Mach. Learn., vol. 11, pp. 7–22, 1993.
[52] M. Kearns, Y. Mansour, "A fast, bottom-up decision tree pruning algorithm with near-optimal generalization," in Proc. 15th Int. Conf. Machine Learning, J. Shavlik, Ed. , pp. 269–277, 1998.
[53] B. Kijsirikul, K. Chongkasemwongse, "Decision tree pruning using backpropagation neural networks", In Proc. Int. Joint Conf. Neural Network(IJCNN), IEEE, Washington DC, USA, Aug. 2002.
[54] F. Esposito, D. Malerba, G. Semeraro, "A comparative analysis of methods for pruning decision trees," IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 5, pp. 476–492, May 1997.