Generalized Entropy for Splitting on Numerical Attributes in Decision Trees

M. Zhong*, M. Georgiopoulos*, G. Anagnostopoulos, M. Mollaghasemi***

*University of Central Florida, Orlando, FL 32816
**Florida Institute of Technology, Melbourne, FL 32791
myzhong@ucf.edu, michaelg@mail.ucf.edu, georgio@fit.edu, mollagha@mail.ucf.edu

Abstract

Decision Trees are well known for their training efficiency and their interpretable knowledge representation. They apply a greedy search and a divide-and-conquer approach to learn patterns. The greedy search is based on the evaluation criterion on the candidate splits at each node. Although research has been performed on various such criteria, there is no significant improvement from the classical split approaches introduced in the early decision tree literature. This paper presents a new evaluation rule to determine candidate splits in decision tree classifiers. The experiments show that this new evaluation rule reduces the size of the resulting tree, while maintaining the tree's accuracy.

Introduction

Decision Tree is a specific type of algorithm for machine learning. One of the notable and earliest decision trees is CART (Classification and Regression Tree) by Breiman et al. (1984). This paper focuses on classification problems only, due to the fact that most techniques in classification trees can be applied to regression trees with minor adjustments.

To learn the examples in a training set with the CART algorithm, a tree is grown in the following process. Initially only one node is generated, with all examples attached to it. The node will be split by a rule based on a single attribute. If the attribute is numerical, the rule is in the form of "is $x_i<b$?" where x_i is the attribute value and b is a threshold; if the attribute is categorical, the rule is in the form of "is $x_i \in$B?" where x_i is the attribute outcome and B is a subset of all the outcomes of x_i among the examples attached to the leaf being split. The CART algorithm selects the best split by enumeration to maximize the split gain, which will be discussed later in detail.

According to the splitting rule, the examples attached to the current node will be divided into two partitions and attached to two new nodes denoted as the children of the current node. The new nodes will also be split, in the same fashion, until the examples in each new node have the same label or cannot be split any more.

The above process is called the growing phase. Usually the resulting tree is over sized and because of that exhibits reduced generalization capability on unseen examples. Hence, "pruning" of this tree is needed, but the details of the tree pruning phase are omitted, because it is not the focus of this paper.

The growing phase of CART is a greedy search based on the split gain measure. Many researchers have tried various measures to improve the resulting tree quality in accuracy and/or size. However, most of these measures are based on the class distribution only and do not consider the attribute distributions, which is important in some applications. As demonstrated in this paper, most of these past splitting techniques fail to work well on some simple problems. In this paper, we propose a new approach considering both the attribute distributions and the class distributions to evaluate the split gain. Experiments demonstrate the advantage of this proposed approach.

The rest of this paper is organized as follows. The section "Related Work" covers the classic definitions of impurity, the split gain and existing variations. The section "Generalized Impurity" derives and explains our new idea. The section "Experiments and Results" describes the comparison between our approach and two others used in two classic algorithms, CART and C4.5 (Quinlan 1993). The section "Conclusion" summarizes this paper.

It is assumed throughout this paper that the reader is familiar with CART and C4.5 decision tree classifiers.

Related Work

In CART, the gain of a split s at node t is defined as the decrease of the impurity

$$Gain(t,s) = Im(t) - \overline{Im(t,s)} \qquad (1)$$

$Im(t)$ is the impurity of a node t, defined as a function of the class proportions:

$$Im(t) = f\left(p_1(t), p_2(t), ..., p_C(t)\right) \qquad (2)$$

where $p_j(t)$ is the proportion of the j-th class in the examples attached to node t for j=1, 2, ... ,C and C is the number of classes. Breiman et al. (1984) pointed out that the function f must satisfy the following conditions:

- f is maximized when $p_1(t) = p_2(t) = ... = p_C(t)$ (most impure)
- f is minimized to zero when only one of the $p_j(t)$'s is one and the rest are zero (completely pure)

- f is concave (which guarantees that the overall impurity after a split will never be larger than before)

The most commonly used impurity functions in CART are the entropy and the Gini Index:

$$Entropy(p_1(t),..., p_C(t)) = -\sum_{j=1}^{C} p_j(t) \log p_j(t) \quad (3)$$

$$Gini(p_1(t),..., p_C(t)) = 1 - \sum_{j=1}^{C} p_j(t)^2 \quad (4)$$

The overall impurity after a split s at node t is defined as:

$$\overline{Im(t,s)} = \frac{N(t^L)Im(t^L) + N(t^R)Im(t^R)}{N(t)} \quad (5)$$

where t^L and t^R are the new nodes formed according to the split and $N(t)$ is the number of examples attached to node t.

Other Split Evaluations

Although prior research has stated that the selection of the impurity function f is not critical as long as it satisfies the conditions specified by Breiman (Breiman et al 1984 and Minger 1989), later experiments revealed that the definition of the split gain, as the only heuristic function used in the greedy search, affects the tree quality in a certain extent (e.g., see Buntine et al 1992). Some of the work reported in the literature and carried out to improve the split evaluations by redefining the split gain, is included below:

- CART provides an option to evaluate the gain using a twoing rule (Breiman et al 1984), which does not directly evaluate gain as the decrease of the impurity, but as the decrease of the impurity in an altered 2-class problem by grouping the class labels into two super classes.
- The well known C4.5 (Quinlan 1993) maximizes the split gain ratio instead of the split gain:

$$GainRatio(t,s) = \frac{Gain(t,s)}{Entropy\left(\frac{N(t^L)}{N(t)}, \frac{N(t^R)}{N(t)}\right)} \quad (6)$$

- Rounds (1980) chooses a splitting threshold b to maximize the Kolmogorov-Smirnov distance assuming two class problem:

$$D(b) = |F(b \mid Y=1) - F(b \mid Y=2)|$$

where $F(x|Y=j)$ is the estimated cumulative distribution function at x given class j. The most advantage of this criterion is that the Kolmogorov-Smirnov distance is independent of the distributions $F(x|Y=j)$. This criterion can also be extended to multi-class problems (Haskell and Noui-Mehidi, 1991).

- De Merckt (1993) maximizes a contrast-entropy ratio:

$$CE(t,s) = \frac{\frac{N(t^L)N(t^R)}{N(t)}\left(m_k^{\,L} - m_k^{\,R}\right)^2}{\overline{Im(t,s)}}$$

where the impurity is the entropy and $m_k^{\,L}$ is the mean value of the attribute used for split among the examples falling to the left partition.

- Taylor (1993) maximizes the MPI (Mean Posterior Improvement):

$$MPI(t,s) = \frac{N(t^L)N(t^R)}{N(t)^2} - \sum_{j=1}^{C} p_j(t) P(L \mid Y=j) P(R \mid Y=j)$$

where $P(L|Y=j)$ means the probability at which an example goes to the left partition given that it has class label j.

- Utgoff, Berkman, and Clouse (1997) applies the direct metrics, which can be the expected number of tests, the minimum description length (MDL), the expected classification cost, or the expected misclassification cost.

It is worth mentioning that some other researchers proposed other approaches for splitting categorical attributes as well. For example, Simovici and Jaroszewicz (2004) applied the Goodman-Kruskal Association Index, and Zhou (1991) utilized the Symmetrical τ Criterion. Although numerical attributes in a finite sized data set can be treated as categorical attributes, the number of their outcomes (distinct values) are usually very large, resulting in impractical time complexity.

Unfortunately none of the above variations improves the performance of the decision tree, in the sense of accuracy or conciseness. Most of the improvements are achieved with small data sets, which weaken their statistical significance. Even worse, these variations do not incorporate the attribute distributions properly enough as to solve some simple problems optimally, as demonstrated in the next section.

Generalized Impurity

Before introducing our approach, let us consider the Cross problem shown below:

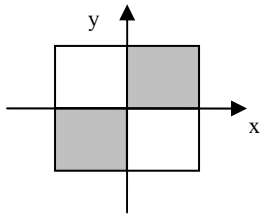

Figure 1: The Cross problem

In the above figure, x and y are the attributes and the two colors represent two classes. The examples are uniformly distributed in the rectangle. Apparently, the optimal splits are "is x<0" and "y<0". However, no matter where the first split is placed (as long as it is univariate), the two partitions still have 50% white area and 50% gray area, which is the same distribution of class labels as before the split. This means the split gain is always zero as long as the gain depends on the class proportions ($p_j(t)$'s) only. Moreover, the projections of the points in each class on either axis completely overlap with each other, resulting in the same probability density function on x and y given any class. Therefore, except De Merckt's and Taylor's approaches, none of the previously mentioned split

approaches can obtain the optimal solution for this problem. Note that De Merckt's and Taylor's approaches happen to yield the optimal solution because they favor central cuts when the impurity cannot be reduced, but they still fail upon a non-symmetric version of the Cross problem.

The most straightforward methods to address this problem are:

- Use non-greedy search (Cantu-Paz, 2003)
- Take the attribute distributions into account for the definition of the impurity, despite that our ultimate goal is to grow a tree with pure leaves (pure in the sense of class labels rather than attributes).

Focusing on the second approach in this paper, we expect that our technique will be able to distinguish the impurity among the following cases of data (see Figure 2):

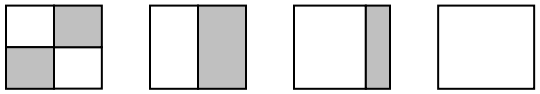

Figure 2: Typical cases of datasets that the proposed technique can efficiently split

It is desired that the impurity decreases from the left most case to the right most case. The main difference between the first (leftmost) case and the second (rightmost) case is that each class in the first case has a wider distribution in the attributes. Most present impurity measures, such as entropy and Gini Index, rely on class proportions only and thus do not distinguish the first case from the second one. On the other hand, it is desired that if a node is pure, the impurity should be zero no matter how widely the attribute scatters. Therefore, it is reasonable to use attribute variance instead of simple frequency count to compute the entropy. To explain this idea more clearly, let us revisit the classic definition of the split gain with the introduction of a weighted impurity:

$$W(t) = N(t)Im(t) \tag{7}$$

It is now easy to see that the split gain can be rewritten as:

$$Gain(t,s) = W(t) - W(t^L) - W(t^R) \tag{8}$$

Equation (8) differs from (1) only by a factor of $N(t)$, which does not affect the selection of the best split.

Let $N_j(t)$ represent the class frequency, namely the number of examples of class j attached to node t.

$$N(t) = \sum_{j=1}^{C} N_j(t) \tag{9}$$

$$p_j(t) = N_j(t)/N(t),\ j = 1,2,...,C \tag{10}$$

Equations (9) and (10) are used to compute *Im(t)* in Equation (2). If we apply the entropy measure as *Im(t)*, it is not difficult to rewrite Equation (3) as:

$$Im(t) = -\sum_{j=1}^{C}\frac{N_j(t)}{N(t)}\log\frac{N_j(t)}{N(t)} = -\frac{1}{N(t)}\sum_{j=1}^{C}N_j(t)\left(\log N_j(t) - \log N(t)\right)$$

$$= \log N(t) - \frac{1}{N(t)}\sum_{j=1}^{C}N_j(t)\log N_j(t)$$

$$W(t) = \left(\sum_{j=1}^{C}N_j(t)\right)\log\left(\sum_{j=1}^{C}N_j(t)\right) - \sum_{j=1}^{C}\left(N_j(t)\log N_j(t)\right) \tag{11}$$

In the above equation, we express $W(t)$ as a function (denoted as *W_Entropy*) of the class frequencies ($N_j(t)$'s). It is not difficult to express $W(t)$ with respect to $N_j(t)$'s using other impurity measures such as the Gini Index.

In order to incorporate the attribute distributions, we now replace Equation (11) with the Generalized Entropy without changing equation (8):

$$W(t) = (1-q)W\_Entropy(N_1(t), N_2(t), ..., N_C(t)) + qW\_Entropy(V_1(t), V_2(t), ..., V_C(t)) \tag{12}$$

$$V_j(t) = \begin{cases} \sum_{d=1}^{D} V_{jd}(t), & N_j(t) > 0 \\ 0, & N_j(t) = 0 \end{cases} \tag{13}$$

$$V_{jd}(t) = \frac{\sum_{i=1}^{N_j(t)}\left(x_{id}^j(t) - \overline{x_d^j(t)}\right)^2}{R_d{}^2} \tag{14}$$

$$\overline{x_d^j(t)} = \frac{\sum_{i=1}^{N_j(t)} x_{id}^j(t)}{N_j(t)} \tag{15}$$

where D is the number of attributes (assumed all numerical for now), $x_{id}^{jt}(t)$ is the value of the d-th attribute in the i-th example of class j attached to node t, and R_d is the range of the d-th attribute within the entire training data set.

In Equation (14), the numerator is the sum of the square distance of the attribute points to the center within the class. We do not use the variance directly because we desire that this quantity be proportional to $N_j(t)$ given the same distribution. The denominator R_d is used for normalization so that our measure will have no bias due to the scaling of the attributes. We assume $R_d>0$ for all attributes; if $R_d=0$, it means the d-th attribute is a constant and it should have been discarded.

In Equation (12), q is a predefined factor between 0 and 1. When $q=0$, the Generalized Entropy reduces to the classic one. We do not recommend $q=1$ because it favors end cuts: if node t is NOT pure but each minor class has exactly one example, then *W_Entropy*($V_1(t)$, $V_2(t)$, ... , $V_C(t)$) is still zero. Our experiments have demonstrated that a good default value for q is $q=0.4$.

Properties

The Generalized Impurity and the corresponding split gain have the following properties, as long as $0<q<1$:

Property A: The Generalized Impurity is non-negative and it is zero if and only if at most one class is present.

Property B: The split gain is always non-negative regardless of the split.

Proof of property A: The function *W_Entropy* is non-negative. If $W(t)$ is zero, *W_Entropy* ($N_j(t)$) must be zero, which means at most one of the $N_j(t)$'s is non-zero; if at most one of the $N_j(t)$'s is non-zero, only the corresponding $V_j(t)$ may be non-zero, which means *W_Entropy* is zero.

Proof of property B: Let $y_{id}^j(t) = x_{id}^j(t)/R_d$.

$$V_{jd}(t) = \sum_{i=1}^{N_j(t)}\left(y_{id}^j(t) - \overline{y_d^j(t)}\right)^2 = \sum_{i=1}^{N_j(t)} y_{id}^j(t)^2 - \frac{\left(\sum_{i=1}^{N_j(t)} y_{id}^j(t)\right)^2}{N_j(t)} \tag{16}$$

$$V_{jd}(t) = \sum_{i=1}^{N_j(t^L)} y_{id}^j(t^L)^2 + \sum_{i=1}^{N_j(t^R)} y_{id}^j(t^R)^2 - \frac{\left(\sum_{i=1}^{N_j(t^L)} y_{id}^j(t^L) + \sum_{i=1}^{N_j(t^R)} y_{id}^j(t^R)\right)^2}{N_j(t^L)+N_j(t^R)}$$

$$= \sum_{i=1}^{N_j(t^L)} y_{id}^j(t^L)^2 - \frac{\left(\sum_{i=1}^{N_j(t^L)} y_{id}^j(t^L)\right)^2}{N_j(t^L)} + \sum_{i=1}^{N_j(t^R)} y_{id}^j(t^R)^2 - \frac{\left(\sum_{i=1}^{N_j(t^R)} y_{id}^j(t^R)\right)^2}{N_j(t^R)}$$

$$+ \frac{\left(N_j(t^L)\sum_{i=1}^{N_j(t^L)} y_{id}^j(t^L) - N_j(t^R)\sum_{i=1}^{N_j(t^R)} y_{id}^j(t^R)\right)^2}{N_j(t^L)N_j(t^R)\left(N_j(t^L)+N_j(t^R)\right)}$$

$$= V_{jd}(t^L) + V_{jd}(t^R) + \frac{\left(N_j(t^L)\sum_{i=1}^{N_j(t^L)} y_{id}^j(t^L) - N_j(t^R)\sum_{i=1}^{N_j(t^R)} y_{id}^j(t^R)\right)^2}{N_j(t^L)N_j(t^R)\left(N_j(t^L)+N_j(t^R)\right)}$$

$$\geq V_{jd}(t^L) + V_{jd}(t^R)$$

Therefore, $V_j(t) \geq V_j(t^L) + V_j(t^R)$. Using the concavity of the entropy function and the monotonicity of *W_Entropy*, we can show that $W(t) \geq W(t^L) + W(t^R)$.

Time Complexity

One of the most important advantages of CART is its low time complexity. In the typical case, the time complexity of the growing phase of CART is $O(DN(\log N)^2)$ where D is the number of attributes and N is the number of training patterns; in the worst case, the complexity is $O(DN^2 \log N)$, which happens only when each split is an end cut (the derivation is omitted in this paper). The typical time complexity requires that when consecutive values of the splitting threshold b are evaluated, the class frequencies ($N_j(t)$'s) should not be counted by going through all the examples attached to the current node, but by the previous class frequencies plus/minus the number of examples switching from the left child to the right one (or inversely).

In our approach, not only $N_j(t)$'s but also $V_j(t)$'s need updating when b is shifted. Nevertheless, this can still be performed in negligible time, because Equation (16) implies that we can update the sum of each attribute and the sum of the square of each attribute as easily as updating $N_j(t)$ to include/exclude a point into/from the left child. Of course, this must be performed for each attribute. For this reason, our approach has a time complexity higher than that of CART by a factor of D, which is usually small.

Application to Categorical attributes

For categorical attributes, we can replace Equation (14) with the weighted entropy function:

$$V_{jd}(t) = W\_Entropy(N_{d1}^j(t), N_{d2}^j(t), \ldots) \qquad (17)$$

where $N_{dk}^j(t)$ is the number of examples attached to node t with class j and the k-th outcome in the d-th attribute. Similarly, we can prove that properties a) and b) still hold true if categorical attributes are present. Equation (11) also allows us to update $V_{jd}(t)$ efficiently in a constant time complexity regardless of the number of examples and the number of outcomes of the d-th attribute. Therefore, our previous analysis of the time complexity is still valid.

Experiments and Results

We implemented and compared the following three evaluations of splitting rules:

- Decrease of Entropy – Equations (8) and (11). It is used in CART.
- Gain Ratio – Equations (6), (8) and (11). It is used in C4.5. For a fair comparison, we did not use C4.5 directly but modified CART with this approach.
- Decrease of Generalized Entropy – Equations (8) and (12). For simplicity, the categorical attributes are not taken into account in (12), although they can be used in a splitting rule. The parameter q was set equal to 0.4. The rest of the algorithm is the same as CART.

We tested these approaches with the Cross problem and some UCI repository problems (Newman et al., 1998). Each data set was processed as following:

1) Randomly shuffle the data in the database
2) Use the first 40% of the data for growing a tree, the next 30% of the data for pruning the tree with the 1-SE rule (Breiman, et al., 1984), and the remaining 30% of the data for testing the tree.

To reduce the randomness factor related to Step 1, we repeated the experiments for 10 times. In each time, we shuffled the data again for a new growing/pruning/testing set, on which we ran all the tested algorithms for a fair comparison. The results shown in Tables 2 and 3 reflect the average performance over the 10 runs.

Data Sets

To ensure statistical significance, we selected only the data sets with 2000 instances or more. For the Cross problem, we generate random points uniformly distributed in the rectangle region $\{(x,y)|-1<x<1,-1<y<1\}$. The class label is set to the sign of $x \cdot y$ for each point. The other data sets are downloaded from the UCI repository. Table 1 lists the statistical information pertinent to the databases.

Name of Database	#Cases	#Numerical attributes	#Nominal attributes	#Classes	Major Class %
Cross	4000	2	0	2	50.22
Abalone	4177	7	1	3	16.4951
Segment	2310	19	0	7	14.2857
Letter	20000	16	0	26	4.065
Waveform	5000	21	0	3	33.92
Pen digits	10992	12	0	10	10.4076
Satellite	6435	36	0	6	23.8228
Opt digits	5620	64	0	10	10.1779
Shuttle	14500	9	0	7	79.1586

Table 1: Statistical information about the databases

Experimental Results

Tables 2 and 3 give the classification results. The results reported there are the mean values out of the 10 runs obtained from the test set (30% of the whole database) which is unseen by the tree. In tables 2 and 3, the second column shows the results of CART, the third column shows those of CART using Gain Ratio, and the last column shows the results of CART using our split gain measure.

Name of Database	Decrease of Entropy	Gain Ratio	Decrease of Generalized Entropy
Cross	99.50%	86.31%	99.96%
Abalone	62.31%	57.58%	62.00%
Segment	94.59%	93.72%	94.49%
Letter	83.86%	82.47%	83.73%
Waveform	76.15%	72.72%	76.30%
Pen digits	94.51%	94.33%	94.54%
Satellite	84.90%	83.84%	84.69%
Opt digits	87.46%	85.64%	86.58%
Shuttle	99.95%	99.95%	99.95%

Table 2: Accuracy of the three approaches

Name of Database	Decrease of Entropy	Gain Ratio	Decrease of Generalized Entropy
Cross	10.4	224	4
Abalone	8.5	196.7	6.1
Segment	20.6	25	20
Letter	956.3	1139.6	928.2
Waveform	33.2	143.6	29.2
Pen digits	112.7	140.2	105.9
Satellite	35.2	84.8	32.2
Opt digits	76.8	101.5	76.5
Shuttle	19.5	28.1	19.3

Table 3: Number of tree leaves for the three approaches

These tables show that the gain ratio is much worse than the gain itself: the accuracy is always worse and the tree is significantly larger. The draw back of C4.5 in the resulting tree sizes has already been pointed out in Lim, Loh, and Shih 2000. Fortunately, C4.5 improves the accuracy by not requiring a pruning set so it can have a larger training set to grow the tree.

Our approach appears better than the criteria used in CART and C4.5, mostly in reducing tree size while achieving similar accuracy. For the Cross problem, our approach always achieved the optimal tree with 4 leaves, while CART's criterion yielded more than twice our tree size and Gain Ratio even resulted in a tree 56 times our tree size. For the benchmark databases, however, the difference in size and accuracy between our approach and the classic ones is much less evident, because the practical problems usually contain redundancy among the attributes, and thus the data is not usually distributed as the data corresponding to the Cross problem. Nevertheless, our approach also reduced the tree size, mostly by 3%-10%, while maintaining the accuracy (the worst deterioration is less than 1%).

Conclusions

In this paper we demonstrated the difficulty of the existing evaluation methods in correctly splitting the data in the growing phase of the tree. Thus, we motivated the reason for a splitting rule that utilizes not only class labels but attribute distributions in the splitting evaluation rules. We introduced such a splitting rule, named Generalized Entropy, which incorporates both the class distribution and the attribute distributions in splitting the data. We proved that our measure has similar properties and comparable time complexity to the classical slitting decision tree measures. Our experiments have also shown that our splitting measure reduces the tree size while maintaining or improving the accuracy. Unfortunately, the improvement is not significant for benchmark databases. Our decision tree splitting method can be readily applied to other classic measurements such as the Gini Index, and can also be generalized to categorical attributes.

Acknowledgment

This work was supported in part by a National Science Foundation (NSF) grant CRCD: 0203446. Georgios Anagnostopoulos and Michael Georgiopoulos acknowledge the partial support from the NSF grant CCLI 0341601.

References

Breiman, L., Friedman, J.H, Olshen, R. A., and Stone, C. J. 1984. *Classification and Regression Trees*. Wadsworth, Belmont CA.

Buntine, W. and Niblett, T. 1992. A Further Comparison of Splitting Rules for Decision Tree Induction. *Machine Learning*, 8:75-85.

Cantu-Paz, E. and Kamath, C. 2003. Inducing Oblique Decision Trees with Evolutionary Algorithms, *IEEE Trans. Evolutionary Computation*, 7(1):54-68.

De Merckt, T. V. 1993. Decision Trees in Numerical Attribute Spaces. In *IJCAI-93*, 1016-1021.

Haskell, R. E. and Noui-Mehidi, A. 1991. Design of Hierarchical Classifiers. In *Proceedings of Computing in the 90's: The First Great Lakes Computer Science Conference*, 118-124, Berlin: Springer-Verlag.

Lim, T. S., Loh, W. Y., and Shih. Y. S. 2000. A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms. *Machine Learning*, 40(3):203-228.

Mingers, J. 1989. An Empirical Comparison of Selection Measures for Decision Tree Induction. *Machine Learning*, 4(2):227-243.

Newman, D. J., Hettich, S., Blake, C. L., and Merz, C. J. 1998. *UCI Repository of machine learning databases*, Department of Information and Computer Science, University of California, Irvine, CA. Available at [http://www.ics.uci.edu/~mlearn/MLRepository.html].

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. San Mateo, Calif.: Morgan Kaufmann

Rounds, E. 1980. A Combined Non-parametric Approach to Feature Selection and Binary Decision Tree Design. *Pattern Recognition*, 12:313-317.

Simovici, D. A. and Jaroszewicz, S. 2004. A Metric Approach to Building Decision Trees Based on Goodman-Kruskal Association Index. *PAKDD*: 181-190

Taylor, P. C. and Silverman, B. W. 1993. Block Diagrams and Splitting Criteria for Classification Trees. *Statistics and Computing*, 3(4):147-161.

Utgoff, P. E. and Clouse, J. A. 1996. A Kolmogorov-Smirno Metric for Decision Tree Induction. *Technical Report* 96-3, University of Massachusetts, Amherst

Zhou, X. J. and Dillon, T. S. 1991. A Statistical-heuristic Feature Selection Criterion for Decision Tree Induction. *IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI* 13(8):834-841.