Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers & Operations Research 31 (2004) 1933–1945
computers & operations research
www.elsevier.com/locate/dsw

# Evaluation of decision trees: a multi-criteria approach

Kweku-Muata Osei-Bryson

*Department of Information Systems and The Information Systems Research Institute, School of Business, Virginia Commonwealth University Richmond, VA 23284, USA*

**Abstract**

Data mining (DM) techniques are being increasingly used in many modern organizations to retrieve valuable knowledge structures from organizational databases, including data warehouses. An important knowledge structure that can result from data mining activities is the decision tree (DT) that is used for the classification of future events. The induction of the decision tree is done using a supervised knowledge discovery process in which prior knowledge regarding classes in the database is used to guide the discovery. The generation of a DT is a relatively easy task but in order to select the most appropriate DT it is necessary for the DM project team to generate and analyze a significant number of DTs based on multiple performance measures. We propose a multi-criteria decision analysis based process that would empower DM project teams to do thorough experimentation and analysis without being overwhelmed by the task of analyzing a significant number of DTs would offer a positive contribution to the DM process. We also offer some new approaches for measuring some of the performance criteria.


*Keywords:* Decision tree; Evaluation; Performance measures; Multi-criteria decision analysis

## 1. Introduction

Data mining (DM) techniques are being increasingly used in many modern organizations to retrieve valuable knowledge structures from organizational databases, including data warehouses. An important knowledge structure that can result from data mining activities is the decision tree (DT) that is used for the classification of future events. The induction of the DT is done using a supervised knowledge discovery process in which prior knowledge regarding classes in the database is used to guide the discovery. The generation of a DT is a relatively easy task but in order to select the most appropriate DT, it is necessary for the DM project team to generate and analyze a

*E-mail address:* kweku.muata@isy.vcu.edu (K.-M. Osei-Bryson).

0305-0548/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0305-0548(03)00156-4

significant number of DTs based on multiple performance measures. Many DM software packages (e.g. C5.0, SAS Enterprise Miner, IBM Intelligent Miner) provide facilities that make the generation of DTs a relatively easy task. However, in using these DM software the DM analyst has to make decisions on various parameter values (e.g. Minimum Number of Cases per Leaf, Splitting Criterion, Minimum Number of Cases for a Split, Maximum Number of Branches from a Node, Maximum Depth of Tree) that could determine the DT that is generated. Even in those situations when some major objectives of the DM project are known (e.g. accuracy of top events in the first quartile), the choice of the parameter values may not be obvious. The DM analyst may thus have to experiment with many different sets of parameter values thus resulting in a significant number different DTs that must be evaluated. Although one may be concerned that the selected DT should give the best performance with regards to Accuracy, there are other criteria (e.g. Simplicity, Stability, Lift) could also be important in determining the most appropriate DT. It should be noted that although the data mining software may generate a single DT as its choice, some software permit the accessing of other DTs without additional computational costs. Most of these DTs would have been rejected during the pruning phase of DT construction when the primary objective was to identify the DT that provided the highest accuracy rate. However, if other performance measures are also important then some of these rejected DTs might actually be worthy of consideration.

The use of multiple performance criteria adds some complexity to the problem of selecting the most appropriate DT. In the case when one of these DTs outperforms all other DTs with regards to all relevant performance measures then the choice of the most appropriate DT is obvious. When this is not the case, he DM analyst and other members of the DM project team would need a good approach for selecting the most appropriate DT given conflicting performance values.

Most previous approaches to comparing decision trees have focused on a single performance measure, typically some measure of accuracy (e.g. [1]), although it is usually acknowledged that multiple factors are important for evaluating DTs (e.g. [2]). Many commercial applications focus on both accuracy and lift (e.g. [3,4]). Han and Kamber [5] in discussing the issue of whether the accuracy criterion is sufficient for evaluating DTs state that “in addition to accuracy, classifiers (DTs) can be compared with respect to their speed, robustness, scalability, and interpretability”. Lim et al. [6] used accuracy, complexity and training time to compare the performance of DT induction algorithms and thus the DTs that they generated. Garofalakis et al. [7] indirectly addressed the issue of accommodating multiple performance measures by preventing the generation of DTs that would violate performance measure constraints. While there is debate about which performance measure is the best (e.g. [1,8]) little attention has been paid to developing a formal approach for comparing DTs that could accommodate multiple performance measures, although DM project teams have to make such comparisons routinely. For as noted by [9] with regards to setting parameter values, there is no “no practicable approach to select ... the most promising combinations early in the process” and as such “it is necessary to experiment with different combinations” but “it is very hard to compare that many models and pick the optimal one reliably”. Thus, it is necessary for the DM project team to generate and analyze a significant number of DTs.

We propose a multi-criteria decision analysis (MCDA) based process to provide support to the DM project team in its effort to select the most appropriate DT. This MCDA based process incorporates subjective opinion about the relative importance of the different performance measures while permitting the team to generate as many DTs as may be necessary to determine the most appropriate DT without being overwhelmed by the information overload that might lead to the consideration

of an insufficient number of DTs. We also offer some new approaches for measuring some of the performance criteria.

## 2. Overview on decision trees

A DT is a tree structure representation of the given decision problem such that each non-leaf node is associated with one of the decision variables, each branch from a non-leaf node is associated with a subset of the values of the corresponding decision variable, and each leaf node is associated with a value of the target (or dependent) variable. There are two main types of DTs: (1) classification trees and (2) regression trees. For a classification tree, the target variable takes its values from a discrete domain, and for each leaf node the DT associates a probability (and in some cases a value) for each class (i.e. value of the target variable). The class that is assigned to a given leaf node of the classification tree results from a form of majority voting in which the winning class is the one that provides the largest class probability even if that probability is less than 50%. In this paper we will focus on the classification tree, which is the most commonly used type of DT, and so henceforth in the paper whenever we use the term decision tree we will be referring to a classification tree.

The generation of a DT involves partitioning the model data set into at least two parts: the training data set and the validation data set (commonly referred to as the test data set). There are two major phases of the DT generation process: the *growth phase* and the *pruning phase* (e.g. [10]). The *growth phase* involves inducting a DT from the training data such that either each leaf node is associated with a single class or further partitioning of the given leaf would result in the number of cases in one or both child nodes being below some specified threshold. The *pruning phase* aims to generalize the DT that was generated in the *growth phase* in order to avoid over fitting. Therefore in this phase, the DT is evaluated against the test (or validation) data set in order to generate a subtree of the DT generated in the *growth phase* that has the lowest error rate against the validation data set. It follows that this DT is not independent of the training data set or the validation data set (i.e. commonly called test data set). For this reason it is important that the distribution of cases in the validation data set correspond to the overall distribution of the cases.

## 3. Overview on performance measures for evaluating decision trees

The most commonly used performance criterion for a DT is the predictive *accuracy rate* (i.e. correct classification rate). For DTs with binary target variables and a specified target event, various combinations of *sensitivity* (*i.e True Positives/Actual Positives*) and *specificity* (*i.e True Negatives/Actual Negatives*) have also been considered as measures of accuracy (e.g. [1]). As noted by Han and Kamber [5], *accuracy rate* is a function of *sensitivity* and *specificity*.

Tree simplicity has also been considered by many researchers. For some, a measure of tree simplicity has been limited to the number of leaves in the DT (e.g. [11]) while others have also suggested that the sizes of the corresponding rules (i.e. number of conjuncts of decision variables) are also relevant, particularly when the rules are to be applied by human beings rather than computers (e.g. [5]). Both of these measures have implications for the interpretability of the DT.

Another measure that could affect the interpretability of the DT is the degree of discriminating power of the leaves. Ideally, one would like to have leaves that are totally pure (i.e. for each leaf all classes except one has zero probability) but that is unlikely occur and so as was previously mentioned the class that is associated with the leaf is simply the class with the largest frequency for the given leaf based on the training data set. However, for a human being a given rule might not be considered to be particularly useful if the probability of the assigned class is less than 50%. In general, for many users of a DT, the rule that is associated with a given leaf is only useful if the probability of the majority class is at least some specified cut-off value $\tau(> 0.50)$. Thus for some situations a *Discriminatory Power* performance measure might also be appropriate for evaluating the performance of the DT. We will define one such measure later in the paper.

The *stability* performance criterion concerns our interest that there should not be much variation in this *predictive accuracy rate* when a DT is applied to different data sets. Thus, at a minimum one might expect that there should not be much variation in predictive accuracy of the DT on the validation data set when compared to that for the training data set. Typically, there is no numeric performance measure for the stability property that is provided by the DT induction algorithm and so often an estimate is made of the stability of the DT based on visual inspection of a lift chart, an approach that is impractical if a large number of DTs are to be compared. For one type of a lift chart, the Percentage Response Chart, the cases are sorted by the predicted probability of the target event (i.e. desired value of the target variable), the cases are grouped into deciles, and the actual probability of each decile is computed. The lift chart thus consists of a set of line segments, such that each line segment corresponds to a leaf of the DT. Instability shows up in the form of a jagged curve, where dips indicate that the accuracy of the given DT is worse than a random guess (e.g. [4]). As noted previously visual inspection of a significant number of DTs in order to assess their stability could be a daunting task.

## 4. Definition of some performance measures

In this section, we will provide an approach to measuring the performance of DTs with regard to some of the performance criteria. The reader should note that we do not claim that this set of performance measures is exhaustive or that each performance measure of this set is relevant in every situation. Rather our objective is to show how values for some of these measures could be determined, often based on data that is normally generated by the DT induction algorithm.

### 4.1. Stability (STAB)

One coarse measure of stability is given by $\mathrm{STAB_C} = \mathrm{Min}\{\mathrm{ACC_T}/\mathrm{ACC_V}, \mathrm{ACC_V}/\mathrm{ACC_T}\}$ where $\mathrm{ACC_V}$, $\mathrm{ACC_T}$ are the accuracy rates for training and validation, respectively. It follows that $\mathrm{STAB_C} \in (0, 1]$, with higher values of $\mathrm{STAB_C}$ indicating higher stability. A finer measure would focus on the relative class frequencies of each leaf based on the validation and training data sets. Given a leaf "$k$": let $\varphi_{\mathrm{V}k}$ be the proportion of the validation data set cases that are associated with this leaf; let $\rho_{\mathrm{V}k}$ be the corresponding posterior probability of the validation data set cases that are positive with regards to the decision event (i.e. target event with the maximum posterior probability for leaf $k$); let $\rho_{\mathrm{T}k}$ be the corresponding proportion of the training data set cases that are positive

with regards to the target event. The stability of leaf $k$ based on the training and validation data sets can be defined as $\sigma_{VTk} = \text{Min}\{\rho_{Vk}/\rho_{Tk},\ \rho_{Tk}/\rho_{Vk}\}$, where $\sigma_{VTk} \in (0, 1]$, with higher values of $\sigma_{VTk}$ indicating higher stability. Given this measure the stability of the DT with regards to its performance on the training and validation data sets can be defined as $\text{STAB}_F = \sum_k \varphi_{Vk}\ \sigma_{VTk}^{\varphi_{Vk}}$, where $\text{STAB}_F \in (0, 1]$, with higher values of $\text{STAB}_F$ indicating higher stability. The reader may note that $\text{STAB}_F$ that is just the weighted sum of the stability of the individual leaves.

### 4.2. Simplicity (SIMPL)

In some situations where the DT is to be used as both an explanatory and predictive model, it is important that the DT should be as simple as possible. Simplicity, or equivalently complexity, is often considered to be a function of the number of leaves in the DT, and the rule length. Below we describe approaches to measuring both.

#### 4.2.1. Simplicity based on number of leaves ($SIMPL_{LEAF}$)

Although it is often stated with all else being approximately equal, the fewer the leaves the better one should include a caveat with that statement as we are often not interested in a DT with only a single leaf and for other situations even a DT with two leaves might not be useful. In other words for different DT problem instances there may be different utility functions that map the number of leaves to the simplicity measure. Let us assume that we have such a function such that the complexity $\text{SIMPL}_{\text{LEAF}} = f_{\text{LEAF}}(|K|)$, where $K$ is the index set of the leaves in the DT, and $f_{\text{LEAF}}(|K|)$, is a non-increasing function such that $\text{SIMPL}_{\text{LEAF}} \in (0, 1]$, with higher values of $\text{SIMPL}_{\text{LEAF}}$ indicating higher simplicity.

#### 4.2.2. Simplicity based on rule size ($SIMPL_{RULE}$)

For a given rule, its length (i.e. the number of conjuncts in the rule) provides a measure of the complexity of the rule then another simplicity measure for the DT could be based on the mean rule length of the rules in the DT. Let $x_k$ be the rule length for rule $k \in K$. The mean rule length of the DT could be defined as $x_{\text{Mean}} = \sum_k \varphi_{Vk} x_k$, which is just the weighted sum of the length of each rule. The corresponding rule length based simplicity measure is defined as $\text{SIMPL}_{\text{RULE}} = f_{\text{RULE}}(x_{\text{Mean}})$, where $f_{\text{RULE}}(x_{\text{Mean}})$ is a non-increasing function such that $\text{SIMPL}_{\text{RULE}} \in (0, 1]$, with higher values of $\text{SIMPL}_{\text{RULE}}$ indicating higher simplicity. We will provide an example of such a function in our illustrative example.

### 4.3. Discriminatory power (DSCPWR)

A measure of the *usefullness* of the DT could be based on the discriminatory power of the leaves, with leaf nodes that have low ambiguity with regards to the class to which a case is to be assigned being more desirable. Recall that we are focusing on DTs that generate the posterior probabilities for each value of the target variable, and that the decision for a given leave is the target event (i.e. value of the target variable) that has the maximum posterior probability for that leaf. For a given predictive modeling problem, let $\tau$ be the cut-off value for the posterior probability such that the user would be comfortable with the decision associated with that leaf only if the posterior probability of the decision event (i.e. target event with the largest posterior probability based on the training data)

was greater than or equal to $\tau$. Let $\Psi(k)=1$ if $\rho_{Tk} \geqslant \tau$; and $\Psi(k)=0$ if $\rho_{Tk} < \tau$, where $\rho_{Tk}$ is the posterior probability of the decision event; and let $\varphi_{Vk}$ be the proportion of the validation cases that are associated with leaf $k$. A fine measure of accuracy could be defined as: DSCPWR$=\sum_k \varphi_{Vk}\Psi(k)$, where DSCPWR$\in[0,1]$, with higher values of DSCPWR indicating higher discriminatory power. The rationale here is that predictions of those leaves whose maximum posterior probability is less than the user defined cut-off values are questionable.

## 5. Formulating the evaluation problem

### 5.1. Approach to ranking the decision trees

Given for each of the DTs we have values for the set of performance measures, the question still remains as to how we will go about selecting the most appropriate DT. Given that we are dealing with multiple performance criteria then our evaluation problem is in fact a multiple criteria decision-making (MCDM) problem. In formal terms, MCDM problems are said to involve the prioritization of a set of alternatives in situations that involve multiple, sometimes conflicting criteria. Various formal techniques have been proposed including the weighing model and outranking methods. In this paper, we will focus on the weighing model because of its popularity, relative simplicity and intuitive appeal.

Each DT has a performance vector $DT_i \boldsymbol{v}_i=(v_{i1}, v_{i2}, \ldots, v_{i|J|})$ where $v_{ij}$ is $DT_i$'s score with regards to performance measure $j$, and $J$ is the index set of the performance measures. An intuitively appealing approach for comparing the overall performance of the DTs would be to compute each DT's composite score as the weighted sum of its performance with regards to the individual measures. Thus for DT "$i$" the composite score would be $s_i=\sum_{j\in J} v_{ij}w_j$, where $w_j$ is the weight of performance "$j$" for the given evaluation problem. Given a pair of DTs and our set of weights, $DT_h$ would be preferable to $DT_i$ if $s_h > s_i$. There are some preference relationships that are independent of the weights. For example $DT_h$ would be said to *dominate* $DT_i$ if $\boldsymbol{v}_h=(v_{h1}, v_{h2}, \ldots, v_{h|J|}) \geqslant \boldsymbol{v}_i=(v_{i1}, v_{i2}, \ldots, v_{i|J|})$ and $DT_h$ outperforms $DT_i$ with respect at least one of the performance measures (i.e. $v_{hj} > v_{ij}$ for at least one $j\in J$). It should be noted that if $DT_h$ *dominates* $DT_i$, then $s_h > s_i$ no matter which set of weights is used. A DT that is not dominated by any other DT is said to be *non-dominated*.

#### 5.1.1. Generating weights

Various approaches are available for generating weights $w_j$ from the subjective inputs of evaluators, both for individual and group decision-making contexts, and for situations when the inputs are precise or imprecise (e.g. [12–15]). The application of those techniques requires estimates of the relative importance of pairs of performance measures, and result in a weight vector that is a synthesis of the input pairwise comparison information. While some of these techniques require that pairwise comparison information be provided for each pair of performance measures, for others it is not necessary that estimates be provided for all pairs of measures (e.g. [13–15]). Given the nature of our evaluation problem, we will assume that initially the evaluator is not certain about the numeric estimate of the pairwise comparisons and as such we will provide for the evaluator to make imprecise numeric estimates in the form of numeric intervals. For situations involving an individual

evaluator, techniques described in [14] can be used to generate the corresponding interval weight vector, while for situations involving a group of evaluators, techniques described in [15] can be used to generate a set of consistent weights the corresponding normalized interval weight vector. Thus, the evaluator specifies the imprecise numeric estimate of the relative importance of performance measure $j$ compared to performance measure $k$ as $a_{jk}=[a_{\mathrm{L}:jk}, a_{\mathrm{U}:jk}]$. The weight generation technique produces: (a) a complete imprecise, consistent pairwise comparison matrix $C=\{c_{jk}=[c_{\mathrm{L}jk}, c_{\mathrm{U}jk}]\}$; and (b) a consistency indicator that provides a measure of the consistency of the input pairwise comparison data. The reader may refer to [14] for details of these techniques.

#### 5.1.2. Linear programming formulation

Based on our MCDM model, determining if DT "$h$" could be the most appropriate DT would involve computing its best score given the set of weights that are consistent with the subjective opinion on the relative importance of the performance measures. This involves solving the following linear programming problem:

$P_{\mathrm{DT}h}$ : Max $s_h$

(1) $\sum_{j\in J} v_{ij}w_j - s_i = 0 \quad \forall i \in \Phi$,
(2) $w_j - c_{\mathrm{L}jk}w_k \geqslant 0 \quad \forall j,\ k \in J,\ j \neq k$,
(3) $w_j - c_{\mathrm{U}jk}w_k \leqslant 0 \quad \forall j,\ k \in J,\ j \neq k$,
(4) $\sum_{j\in J} w_j = 1$
(5) $w_j \geqslant 0 \quad \forall j \in J$,

where $J$ is the set of performance measures; constraint 1 is used to compute the score (i.e. $s_i$) for each $\mathrm{DT}_i$ as a weighted sum of the relevant performance measures; constraints 2 and 3 ensure that the set of weights that are used to compute the scores of the DTs is consistent with the subjective opinion on the relative importance of the performance measures (i.e. $(w_j/w_k) \geqslant c_{\mathrm{L}jk}$; $(w_j/w_k) \leqslant c_{\mathrm{U}jk}$); and constraints 4 and 5 ensure that the weights are non-negative and normalized to sum to 1, thus ensuring that each $s_i \in [0, 1]$.

### 5.2. Description of the procedure for evaluating decision trees

Let $\Omega_{\mathrm{SPLTCRT}}$ be the set of selected splitting methods (e.g. Entropy, Chi-Square, Gini), $\Omega_{\mathrm{MINLF}}$ be the set of selected values for the Minimum Number of Cases per Leaf, $\Omega_{\mathrm{MINSPLT}}$ be the set of selected values for the Minimum Number of Cases for a Split, $\Omega_{\mathrm{MAXBRN}}$ be the set of selected values for the Maximum Number of Branches from a Node, $\Omega_{\mathrm{MAXDPTH}}$ be the set of selected values for the Maximum Depth of Tree.

*Step* 1: Preparation

(a) Specify the set of performance measures $J$.
(b) Specify the sets $\Omega_{\mathrm{SPLTCRT}}$, $\Omega_{\mathrm{MINLF}}$, $\Omega_{\mathrm{MINSPLT}}$, $\Omega_{\mathrm{MAXBRN}}$, $\Omega_{\mathrm{MAXDPTH}}$.
(c) If Discriminatory Power (DSCPWR) is one of the selected performance measure, specify the cut-off value $\tau$ for leaf ambiguity.
(d) Specify threshold values for accuracy $\xi_{\mathrm{ACC\_V}}$, stability $\xi_{\mathrm{STAB}}$, and any other performance measure.

(e) Specify the utility function for Simplicity based on the Number of Leaves, and the utility function for Simplicity based on the Chain Lengths of the Rules.

*Step* 2: Generate weights for performance measures

(a) The evaluator(s) from the DM project team specify numeric pairwise comparison data on relevant importance of pairs of performance measures. It is not necessary that a pairwise comparison entry be made for each pair of performance measures but each performance measure must be included in at least one pairwise comparison.
(b) Generate the corresponding normalized weight vector and consistency indicator using a weight vector generation technique (e.g. [14,15]).
(c) If the consistency indicator value is acceptable then go to Step 3, otherwise repeat Step 2.

*Step* 3: Generate candidate decision trees and compute performance measures

For each combination of parameter values from the sets $\Omega_{\text{SPLTCRT}}$, $\Omega_{\text{MINLF}}$, $\Omega_{\text{MINSPLT}}$, $\Omega_{\text{MAXBRN}}$, $\Omega_{\text{MAXDPTH}}$, generate the corresponding $\text{DT}_i$ and calculate the performance measures (e.g. $\text{ACC\_V}_i$, $\text{STAB}_i$).

Let $\Phi$ be the set of all DTs that were generated in this step.

*Step* 4: Determine set of relevant decision trees

(a) Exclude DTs which violate any of the threshold values for the performance criteria from $\Phi$.
(b) Identify and exclude *dominated* DTs from $\Phi$. At this step $\Phi$ now only contains those non-dominated DTs that satisfy all threshold constraints.

*Step* 5: Determine the 'most appropriate' decision tree

(a) Let $\Phi$ be the set of non-dominated DTs that satisfy all threshold constraints. Formulate and solve problem $P_{\text{DT}h}$ for each $h \in \Phi$.
(b) Order the DTs in $\Phi$ in descending sequence based on their values of $s_h$.

## 6. Illustrative example

Our illustrative example involves ten decision trees (DT01–DT10) that were generated by the SAS Enterprise Miner from the HMEQ data set. Table 1 displays data from the first two DTs (DT01, DT02) that will be used to generate values for the performance measures which we will assume to be: Validation Classification Rate (ACC_V), Fine Stability ($\text{STAB}_\text{F}$), discriminatory power on the ambiguity of the leaves (i.e. DSCPWR), simplicity based on the number of leaves ($\text{SIMPL}_{\text{LEAF}}$), simplicity based on the rule lengths ($\text{SIMPL}_{\text{RULE}}$).

*Step* 1: Preparation

In discussion with the end-users the DM analyst was able to determine that: (a) the cut-off value for leaf ambiguity be $\tau = 0.75$; (b) the threshold value for validation accuracy was $\xi_{\text{ACC}} = 0.80$; the threshold value for discriminatory power was $\xi_{\text{DSCPWR}} = 0.75$, and the threshold value for stability was $\xi_{\text{STAB}} = 0.90$; (c) if a DT did not have a positive value for $\text{SIMPL}_{\text{LEAF}}$ then it should be excluded.

Table 1
Data on first two DTs

| | DT01 | | | | DT02 | | | |
|---|---|---|---|---|---|---|---|---|
| | Classification rate: 0.885; Number of leaves: 25 | | | | Classification rate: 0.860; Number of leaves: 7 | | | |
| LEAF ID | Training proportion | Validation frequency | Chain proportion | Validation cases | Training proportion | Validation frequency | Chain proportion | Validation cases |
| 1 | 0.792 | 0.917 | 5 | 12 | 0.620 | 0.617 | 1 | 433 |
| 2 | 0.600 | 0.429 | 5 | 7 | 0.957 | 0.931 | 1 | 743 |
| 3 | 0.923 | 0.833 | 5 | 6 | 0.932 | 0.944 | 2 | 642 |
| 4 | 0.773 | 0.718 | 4 | 39 | 0.829 | 0.870 | 2 | 100 |
| 5 | 0.900 | 0.600 | 4 | 5 | 0.885 | 0.909 | 2 | 11 |
| 6 | 0.960 | 0.583 | 4 | 12 | 0.722 | 0.364 | 2 | 11 |
| 7 | 0.850 | 0.846 | 4 | 13 | 0.964 | 0.963 | 1 | 27 |
| 8 | 0.733 | 0.667 | 4 | 6 | | | | |
| 9 | 0.609 | 0.481 | 5 | 27 | | | | |
| 10 | 0.917 | 0.875 | 5 | 8 | | | | |
| 11 | 0.655 | 0.308 | 4 | 13 | | | | |
| 12 | 0.813 | 1.000 | 4 | 1 | | | | |
| 13 | 0.656 | 0.654 | 5 | 26 | | | | |
| 14 | 0.818 | 0.500 | 5 | 2 | | | | |
| 15 | 0.792 | 0.714 | 4 | 35 | | | | |
| 16 | 0.615 | 0.800 | 5 | 10 | | | | |
| 17 | 0.895 | 0.800 | 5 | 5 | | | | |
| 18 | 0.636 | 0.556 | 5 | 9 | | | | |
| 19 | 0.740 | 0.625 | 3 | 32 | | | | |
| 20 | 0.730 | 0.824 | 2 | 125 | | | | |
| 21 | 0.937 | 0.794 | 2 | 34 | | | | |
| 22 | 1.000 | 1.000 | 2 | 6 | | | | |
| 23 | 0.957 | 0.931 | 1 | 743 | | | | |
| 24 | 0.908 | 0.929 | 1 | 764 | | | | |
| 25 | 0.964 | 0.963 | 1 | 27 | | | | |

The DM analyst was also able to determine from these discussions that for the end-users the ideal DT would have between four through eight leaves; that a DT with less than two leaves or more than twenty leaves was unacceptable; and that value of other acceptable DTs would be based on how well they compared with an ideal DT with regard to the number of leaves. Based on this information the utility function for the simplicity based on the number of leaves was defined as follows:

$$f_{\mathrm{LEAF}}(|K|) = 0 \quad \text{if } |K| < 2 \text{ or } |K| > 20,$$

$$f_{\mathrm{LEAF}}(|K|) = (4 - |K|)/(4 - 2) \quad \text{if } 2 \leqslant |K| < 4,$$

$$f_{\mathrm{LEAF}}(|K|) = 1 \quad \text{if } 4 \leqslant |K| \leqslant 8,$$

$$f_{\mathrm{LEAF}}(|K|) = (20 - |K|)/(20 - 8) \quad \text{if } 8 < |K| \leqslant 20.$$

Table 2
Pairwise comparisons of relative importance of performance measures

| | ACC_V | DSCPWR | $STAB_F$ | $SIMPL_{RULE}$ | $SIMPL_{LEAF}$ |
|---|---|---|---|---|---|
| ACC_V | | | I: [0.80, 1.00]<br>O: [0.80, 1.00] | | |
| DSCPWR | I: [0.80, 1.00]<br>O: [0.80, 1.00] | | I:<br>O: [0.77, 0.83] | | |
| $STAB_F$ | | | | | |
| $SIMPL_{RULE}$ | I: [0.50, 0.80]<br>O: [0.52, 0.77] | I: [0.50, 0.80]<br>O: [0.63, 0.80] | I: [0.50, 0.80]<br>O: [0.50, 0.58] | | I: [0.90,1.11]<br>O: [0.90,1.11] |
| $SIMPL_{LEAF}$ | I:<br>O: [0.58, 0.69] | I:<br>O: [0.69, 0.72] | I:<br>O: [0.56, 0.58] | I: [0.90,1.11]<br>O: [0.90,1.11] | |

I: input entries; O: consistent output entries.

The DM analyst was able to determine that the ideal DT would have an average chain length ($x_{Mean}$) that was no greater than 2; that a DT with an average chain length ($x_{Mean}$) more than 6 was unacceptable; and that value of other acceptable DTs would be based on how well they compared with an ideal DT with regard to the average chain length. Based on this information the utility function for the simplicity based on the average chain length was defined as follows:

$$f_{RULE}(x_{Mean}) = 1 \quad \text{if } x_{Mean} \leqslant 2,$$

$$f_{RULE}(x_{Mean}) = (6 - x_{Mean})/(6 - 2) \quad \text{if } 2 < x_{Mean} \leqslant 6,$$

$$f_{RULE}(x_{Mean}) = 0 \quad \text{if } x_{Mean} > 6.$$

*Step* 2: Generate weights for performance measures

The DM project team did pairwise comparisons on the relative importance of the 5 performance measures. Table 2 displays in the Input Pairwise comparison data that was provided and the output consistent pairwise comparison data that was generated by the Inner Interval Weight Generation Procedure [14]. The reader may observe that pairwise comparisons were not offered for each possible pair of performance measures but that consistent output pairwise comparison data was generated for each distinct pair of performance measures by Inner Interval Weight Generation Procedure.

*Step* 3: Generate candidate decision trees and compute performance measures

Using formulas for calculating each performance measure, and data collected on each of the 10 DTs that are similar for that displayed in Table 1, the values of our performance measures were generated for the 10 DTs. Table 3 displays a list of the DTs ordered in descending sequence based on validation Classification Accuracy (i.e. ACC_V).

*Step* 4: Determine set of relevant decision trees

We can see that all our DTs satisfy the thresholds for validation accuracy, and ambiguity, stability, and rule simplicity but that one of them (i.e. DT01) violates the threshold for $SIMPL_{LEAF}$ although it is one of the top three DTs based on accuracy. Further DT03 is dominated by DT04 since DT03 never outperforms DT04 on any performance measure while DT04's outperforms DT03 for some of

Table 3
Performance measures for DTs ordered by validation accuracy

| DT# | ACC_V | DSCPWR | $STAB_F$ | $SIMPL_{RULE}$ | $SIMPL_{LEAF}$ |
|---|---|---|---|---|---|
| 07 | 0.888 | 0.799 | 0.995 | 0.930 | 0.417 |
| 09 | 0.887 | 0.816 | 0.976 | 1.000 | 0.750 |
| 01 | 0.885 | 0.870 | 0.951 | 1.000 | 0.000 |
| 10 | 0.885 | 0.800 | 0.975 | 1.000 | 1.000 |
| 06 | 0.884 | 0.800 | 0.978 | 1.000 | 0.917 |
| 04 | 0.883 | 0.810 | 0.972 | 1.000 | 0.833 |
| 03 | 0.881 | 0.800 | 0.961 | 1.000 | 0.833 |
| 08 | 0.862 | 0.780 | 0.997 | 1.000 | 1.000 |
| 02 | 0.860 | 0.774 | 0.979 | 1.000 | 1.000 |
| 05 | 0.833 | 0.800 | 0.984 | 1.000 | 1.000 |

Table 4
Non-dominated DTs ordered by validation accuracy

| DT# | ACC_V | DSCPWR | $STAB_F$ | $SIMPL_{RULE}$ | $SIMPL_{LEAF}$ |
|---|---|---|---|---|---|
| 07 | 0.888 | 0.799 | 0.995 | 0.930 | 0.417 |
| 09 | 0.887 | 0.816 | 0.976 | 1.000 | 0.750 |
| 10 | 0.885 | 0.800 | 0.975 | 1.000 | 1.000 |
| 06 | 0.884 | 0.800 | 0.978 | 1.000 | 0.917 |
| 04 | 0.883 | 0.810 | 0.972 | 1.000 | 0.833 |
| 08 | 0.862 | 0.780 | 0.997 | 1.000 | 1.000 |
| 05 | 0.833 | 0.800 | 0.984 | 1.000 | 1.000 |

the performance measures (i.e. ACC_V, DSCPWR, $STAB_F$). Similarly DT08 dominates DT02. We can therefore exclude DT01, DT03 and DT02 from further consideration.

Table 4 displays the set of non-dominated DTs that satisfy the threshold constraints.

- Since *DT07*, *DT04* and *DT08* provide the best values for ACC_V, DSCPWR, and $STAB_F$, respectively, then they are not dominated by any other DT.
- *DT09* provides the second highest value of ACC_V to DT07 but DT09 outperforms DT07 with regards to DSCPWR, $SIMPL_{RULE}$, and $SIMPL_{LEAF}$. It follows that DT09 is non-dominated.
- *DT10* provides the third highest value of ACC_V (after DT07, DT09) but DT10 outperforms both DT07 and DT09 with regards to $SIMPL_{LEAF}$. It follows that DT10 is non-dominated.
- *DT06* could only be dominated by DT07, DT09, and DT10 but DT06 outperforms DT07 and with regards to $SIMPL_{LEAF}$, DT06 outperforms DT09 and with regards to $STAB_F$ and $SIMPL_{LEAF}$, and DT06 outperforms DT10 and with regards to $STAB_F$. It follows that DT06 is also non-dominated.
- *DT05* provides the third highest value of $STAB_F$ but DT05 outperforms DT07 and DT08 with regards to ambiguity of leaves (DSCPWR). It follows that DT05 is also non-dominated.

*Step* 5: Determine 'most appropriate' decision tree

Problem $P_{DTh}$ was formulated and solved for each our non-dominated DTs that do not violate any of the threshold constraints. Table 5 displays the ranking of the DTs and for each DT its optimal

Table 5
Composite performance scores for non-dominated DTs

| DT# | $s_i$ | Best weight vector (ACC_V, DSCPWR, $STAB_F$, $SIMPL_{RULE}$, $SIMPL_{LEAF}$) |
|---|---|---|
| 10 | 0.927 | (0.221016, 0.205590, 0.266447, 0.153473, 0.153473) |
| 8 | 0.923 | (0.216054, 0.208354, 0.270028, 0.155536, 0.150028) |
| 5 | 0.918 | (0.216054, 0.208354, 0.270028, 0.155536, 0.150028) |
| 6 | 0.915 | (0.216054, 0.208354, 0.270028, 0.155536, 0.150028) |
| 4 | 0.903 | (0.216054, 0.208354, 0.270028, 0.155536, 0.150028) |
| 9 | 0.893 | (0.216054, 0.208354, 0.270028, 0.155536, 0.150028) |
| 7 | 0.836 | (0.249385, 0.199508, 0.258542, 0.148920, 0.143646) |

weight vector, where the optimal weight vector is consistent with the output consistent pairwise comparison data generated in Step 2. We can see that DT10 is the top ranked DT while DT07 is the lowest ranked DT even though DT07 had the highest accuracy rate. It should be noted that given the set of possible weights that even if DT07 had accuracy rate (ACC_V) of 0.95 and discriminatory power (DSCPWR) of 0.90 but its values for all other performance measures were unchanged that its best composite score would be 0.872 and so it still would not be the top ranked DT.

The reader in observing that for each of these DTs the weight for stability is greater than that of accuracy, may raise the point that it is unreasonable to expect that stability would be considered more important than accuracy by the project team. However, it should be noted that given the fact that the accuracy rate of each DT surpassed the specified threshold then stability would be considered to be more important than accuracy but that improvement in accuracy above the threshold was still important.

## 7. Conclusion

In this paper, we have investigated the problem of selecting the most appropriate decision tree (DT), and have presented an MCDA based process that could be used to provide support to the DM project team that are faced with the task of selecting the most appropriate DT given the need to accommodate all significant performance measures in their selection and experimentation process. We have also provided more sophisticated approaches for measuring DT performance with regard to some of the performance criteria (i.e. stability, discriminating power, simplicity). Given the increasing use and importance of DT induction, a technique that would empower DM project teams to do thorough experimentation and analysis without being overwhelmed by the task of analyzing a significant number of DTs would offer a positive contribution to the DM process.

## References

[1] Bradley A. The use of area under ROC curve in the evaluation of machine learning algorithms. Pattern Recognition Letters 1997;30(7):1145–59.
[2] Bohanec M, Bratko I. Trading accuracy for simplicity in decision trees. Machine Learning 1994;15:223–50.

[3] Piatetsky-Shapiro G, Steingold S. Measuring lift quality in database marketing. SIGKDD Explorations 2001;2(2): 76–80.
[4] Berry M, Linoff G. Mastering data mining: the art and science of customer relationship management. New York, NY: Wiley, 2000.
[5] Han J, Kamber M. Data mining: concepts and techniques. New York, NY: Morgan Kaufman, 2001.
[6] Lim T-S, Loh W-Y, Shih Y-S. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning 2000;40:203–28.
[7] Garofalakis M, Hyun D, Rastogi R, Shim K. Efficient Algorithms for Constructing Decision Trees with Constraints. Proceedings of the 6th ACM SIGKDD International Conference on Data Mining and Knowledge Discovery (KDD-2000), Boston, MA, 2000. p. 335–9.
[8] Provost F, Fawcett T, Kohavi R. The Case Against Accuracy Estimation for Comparing Induction Algorithms. In: Shavlik J, editor. Proceedings of the Fifteenth International Conference on Machine Learning (ICML98), San Francisco, CA: Morgan Kaufmann, 1998. p. 445–53.
[9] Gersten W, Wirth R, Arndt D. Predictive Modeling in Automotive Direct Marketing: Tools, Experiences and Open Issues. Proceedings of the 6th ACM SIGKDD International Conference on Data Mining and Knowledge Discovery (KDD-2000), Boston, MA, 2000. p. 398–406.
[10] Kim H, Koehler G. Theory and practice of decision tree induction. Omega 1995;23(6):637–52.
[11] Esposito F, Malerba D, Semeraro G. A Comparative Analysis of Methods for Pruning Decision Trees. IEEE Transactions on Pattern Analysis and Machine Intelligence 1997;19:476–91.
[12] Saaty T. The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation. New York: McGraw-Hill, 1980.
[13] Bryson N. A Goal Programming for Generating Priority Vectors. Journal of the Operational Research Society 1995;46:641–8.
[14] Bryson N, Mobolurin A, Ngwenyama O. Modelling pairwise comparisons on ratio scales. European Journal of Operational Research 1995;83:639–54.
[15] Bryson N, Joseph A. Generating consensus priority interval vectors for group decision making in the AHP. Journal of Multi-Criteria Decision Analysis 2000;9(4):127–37.