Constructing Decision Trees with Multiple Response Variables

Seong-Jun Kim [1], Kang Bae Lee [2]

[1] Kangnung National University, Department of Industrial Engineering (sjkim@kangnung.ac.kr)
[2] Catholic University of Pusan, School of Business Administrations (kblee@cup.ac.kr)

Abstract

Data mining is a process of discovering meaningful patterns in large data sets that are useful for decision making and has recently received an amount of attention in a wide range of business and engineering fields. Decision tree, also known as recursive partitioning or rule induction, is one of the most frequently used methods for data mining. A decision tree, on a divide-and-conquer basis, provides a set of rules for classifying samples in the learning data set. Most of works on decision tree have been conducted for the case of single response variable. However situations where multiple response variables should be considered arise from many applications, for example, manufacturing process monitoring, customer management, and clinical and health analysis. This article concerns constructing decision trees when there are two or more response variables in the data set. In this article, we investigate node homogeneity criteria such as entropy and Gini index and then present three approaches to constructing decision trees with multiple response variables. To do so, we first describe extensions of entropy and Gini index to the case in which multiple response variables are of concern. A weighting method for node splitting is also explained. Next, we present a decision tree minimizing an expected loss due to misclassifications. To illustrate the procedures, numerical examples are given with discussions.

1. Introduction

Widespread use of network and information technologies has made it easier to collect data and created large databases. Accordingly, the role of data mining becomes more important. Data mining is the process of discovering interesting patterns in databases that are helpful to decision making[1]. Data mining has been used for practical applications in a variety of domains of biomedical, business, and industrial fields[2]. Two primary methodologies for data mining are machine learning and statistical analysis. Machine learning is the study of computational methods to automate the process of knowledge acquisition from examples[3]. It is known that, compared with parametric statistical analyses, machine learning techniques are more suitable for data mining of a large complex data set. This is because such data set is likely to be under high dimensionality, multicollinearity and non-homogeneity[4]. Major categories of machine learning techniques are decision trees, neural networks, case-based reasoning, and genetic algorithms[1]. According to the recent study by Bose and Mahapatra[1], using decision tree is most popular in the data mining of business fields. They reported that about a half of data mining applications is based upon decision tree methods. For more details, see Bose and Mahapatra[1].

Decision tree, also known as recursive partitioning, provides a set of rules for classifying samples in the data set on a divide-and-conquer basis. Decision trees are broadly divided into classification and regression trees. Decision tree is called classification tree when the response variable is categorical, and regression tree when the response variable is numerical. One of the most famous works on decision trees is probably Classification and Regression Trees(CART) by Breiman et al.[5]. In CART, a node is split into two offspring nodes. So it is called a binary tree. CART method first finds the maximal tree by a splitting procedure and then the right sized tree by a pruning procedure. Also CHAID, ID3, and C4.5[4, 6] are well-known methods for building decision trees. Although these tree-based techniques have been widely used for data mining especially for automated classification, their applications are mainly restricted to the case in which data set has a single response variable(SRV). However, it is not so difficult to find problems that multiple response variables(MRV) should be studied. For example, Zhang[7] has dealt with 22 explanatory variables and 6 binary responses to analyze building-related occupant complaint syndrome(BROCS) in his clinical research. He has presented a maximized log-likelihood (as a generalization of entropy) and a Hotelling T^2 type statistic for MRV-node splitting. Siciliano and Mola[8] has also used 9 explanatory variables and 3 categorical responses to model the family budget of bank customers. They have proposed a predictability index, based upon Gini index, as an MRV-node splitting

criterion. In manufacturing processes, multiple responses are routinely observed as well. It will be interesting to study decision trees with MRV for process monitoring and diagnosis.

This article deals with decision trees when the data set has two or more response variables, all of which are assumed to be categorical in this work. The purpose of this paper is to present node splitting methods taking into account multiple response variables. The remaining part of this paper is organized as follows. Section 2 outlines decision trees and describes measures of node homogeneity such as entropy and Gini index. Although decision tree has some advantages over other machine learning techniques, there are still problems to solve. Finding a way to accommodate multiple response variables would be one of them, and it is dealt with in Section 3. We actually present three approaches to producing an MRV-decision tree and illustrate node splitting procedures by examples. In Section 4, a summary of our work is given with discussions. Limitations as well as future directions of this research are also stated.

2. An Overview of Decision Trees

2.1 Decision Tree as a Machine Learning(ML) Technique

As mentioned earlier, decision tree is one of ML methods to data mining. According to conditions of explanatory variables (or attributes), whole learning samples are modeled into a decision tree. The resultant tree eventually provides a set of rules for classifying these learning samples. This is the why decision tree based classification is called decision tree induction[2] or rule induction[8]. Modeling result by decision tree is easy to explain, and therefore it is widely used to find rules for classifying new samples. Decision tree is also viewed as one of the supervised learning methods because it is built from learning samples with a known classification. The result of learning is represented as a tree, the nodes of which specify attributes and the branches specify attribute values. Figure 1 shows a hypothetical decision tree.

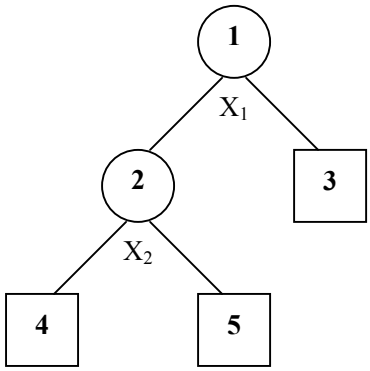

Figure 1. A hypothetical example of decision tree

There are two kinds of nodes in a decision tree; internal nodes(1, 2) and terminal nodes(3, 4, 5). Terminal nodes are also called leaves. In the literature of decision trees, internal node is denoted by the circle, and terminal node by the box. Each of all nodes corresponds a subset of the entire learning set. The root node(1) on the top represents all samples in the learning set. These samples are divided into two disjoint subgroups by the explanatory variable X_1. This process of variable selection and node splitting is continued until each terminal node represents a different class of samples. Eventually, all terminal nodes(3, 4, 5) constitute mutually exclusive and exhaustive subsets of the entire learning set. The resulting decision tree is then applied to a testing set of samples to evaluate its accuracy in classifying new samples. To improve the classification capability of decision tree, it is first important to choose an appropriate node splitting criterion so as to maximize homogeneity of the offspring nodes. Overfitting decision tree to the learning set often drops its classification performance to new samples. In such cases, tree pruning is required to mitigate overfitting before the tree deployed in a real life application. Cross validations can be also used to prevent a decision tree from depending on a specific data set.

2.2 Node Splitting Criteria

As described above, we need a node splitting criterion in order to measure node impurity and to grow the decision tree. Splitting node has to be undertaken such that node impurity can be minimized (or node homogeneity can be maximized). Entropy and Gini index are mainly used for measuring node homogeneity where the data set has a categorical response variable. Consider the following figure to see how to split node.

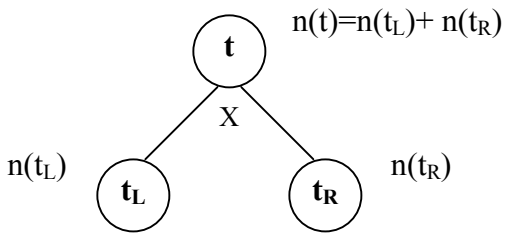

Figure 2. An illustration of node splitting

This figure shows a hypothetical situation that node t is split into two nodes t_L and t_R according to an attribute X. In the figure n(t), $n(t_L)$ and $n(t_R)$ respectively denote the numbers of learning samples in nodes t, t_L, and t_R. Supposing that the response variable Y has K ordered categories, we can obtain the following frequency table which shows that n(t) learning samples are divided into two subgroups.

Table 1. A node splitting result

Node	Y				Sum
	1	2	…	K	
t_L	$n_1(t_L)$	$n_2(t_L)$	…	$n_K(t_L)$	$n(t_L)$
t_R	$n_1(t_R)$	$n_2(t_R)$	…	$n_K(t_R)$	$n(t_R)$
t	$n_1(t)$	$n_2(t)$	…	$n_K(t)$	n(t)

In the table $n_j(t)$, $n_j(t_L)$ and $n_j(t_R)$ are the numbers of learning samples which belong to class j at nodes t, t_L, and t_R respectively, and for j=1,2,…,K,

$$n_j(t_L) + n_j(t_R) = n_j(t)$$

Then an entropy to measure node homogeneity at node t can be expressed as

$$h(t) = -\sum_{j=1}^{K} \frac{n_j(t)}{n(t)} \log \frac{n_j(t)}{n(t)}$$

Note that the smaller entropy, the higher homogeneity (or a lower impurity). One can use Gini index instead, as a node homogeneity measure, as the following.

$$h(t) = 1 - \sum_{j=1}^{K} \frac{n_j^2(t)}{n^2(t)}$$

The amount of homogeneity gain achieved by splitting node t can be then obtained by

$$\eta(t) = h(t) - \frac{n(t_L)}{n} h(t_L) - \frac{n(t_R)}{n} h(t_R) \tag{1}$$

Thus a tree grows by choosing a split that maximizes (1) at each node. Such node splitting is subsequently continued until a stopping rule is satisfied. For example, when η(t) is smaller than a predetermined value, we stop splitting node t and declare it as a terminal one.

2.3 Discussions

Recently Bose and Mahapatra[1] compared several machine learning(ML) techniques for business data mining. According to their study, decision tree has some practical advantages. Especially, in terms of both explanation capability and applicability to large data sets, decision tree method outperforms other ML techniques such as neural network and genetic programming. Ease of operation should be also mentioned as one of the strengths of tree-based methods. In these reasons, it is expected that applying decision trees will become widespread at a variety of domains.

However, there are still problems to cope with in decision tree methods. For example, when the learning set has some irrelevant samples, decision tree tends to divide nodes having few samples and then the resultant tree tends to be too large and overspecified[5]. This leads to a learning unstability, and its classification accuracy becomes under question. To avoid such drawback, tree pruning can be considered. Another limitation lies in that node splitting procedure depends on a single attribute variable. This implies that, as pointed out by Brown et al.[9], a standard decision tree technique is likely to suffer from multi-modal problems. In order to overcome this difficulty, they proposed a multivariate node splitting based upon linear combinations of attributes. Other than these, one of important issues on decision tree is concerning its extension to the case of multiple response variables(MRV). As stated earlier, Zhang[7] included 6 response variables in his clinical study and, however, their work is restricted to binary responses like yes or no. Siciliano and Mola[8] also dealt with 3 response variables in their research. They used a weight sum of Gini indices computed from respective response variables. But how to consider variable importances is not studied. Investigating such restrictions of Zhang[7] and Siciliano and Mola[8], Section 3 presents three node splitting approaches for MRV-situations.

3. Node Splitting Criteria with Multiple Response Variables(MRV)

As stated in Section 1, MRV-situations are often observed in biomedical, business, and other industrial fields. Although it is containing relatively more information about hidden relationships, MRV-data set is difficult to deal with owing to problem complexity and computational burden. Thus, machine learning with MRV will be one of topics worthwhile to study further. As done in SRV-decision trees, of primary importance resides in how node homogeneity should be quantified with MRV. To answer the question, this paper attempts to consider three node splitting schemes. The first deals with extensions of entropy and Gini index. This is done by finding joint frequency distribution of response variables at each node. The covariance structures of joint distributions are also considered for node splitting. The second one is to use a weight sum of node homogeneities for respective response variables. As done in Siciliano and Mola[8], Gini index is used to evaluate node homogeneity. The third approach is concerned with minimizing an expected loss as a node splitting criterion.

3.1 MRV Extensions of Entropy and Gini Index

Response variables are denoted by $Y_1, Y_2, \ldots, Y_M$ where M is the number of response variables in the data set. And suppose that response variable Y_g has K_g ordered classes for g=1,2,…,M. Then a multivariate entropy to represent homogeneity of node t can be defined by

$$h(t) = -\sum_{j_1=1}^{K_1}\sum_{j_2=1}^{K_2}\cdots\sum_{j_M=1}^{K_M}\frac{n_{j_1,j_2,\cdots,j_M}(t)}{n(t)}\log\frac{n_{j_1,j_2,\cdots,j_M}(t)}{n(t)} \tag{2}$$

where $n_{j1,j2,\ldots,jM}(t)$ denotes the number of learning samples whose response values are respectively $Y_1=j_1$, $Y_2=j_2$, …, $Y_M=j_M$ at node t. Similarly, a multivariate Gini index is defined by

$$h(t) = 1-\sum_{j_1=1}^{K_1}\sum_{j_2=1}^{K_2}\cdots\sum_{j_M=1}^{K_M}\frac{n^2_{j_1,j_2,\cdots,j_M}(t)}{n^2(t)} \tag{3}$$

When there are two or more response variables, covariance matrix is obtained. This matrix shows correlations between response variables, which are concerned with node homogeneity in MRV-situations. Supposing that V(t) denote a sample covariance matrix in node t, we can use the determinant of V(t) as an aggregate measure of homogeneity and correlation. That is,

$$h(t) = |V(t)| \tag{4}$$

Pointing out that |V(t)| can be interpreted as Gini index using a single binary response, Zhang[7] included |V(t)| into his criteria for node splitting. In addition, he recommended to use a Hotelling T^2 type statistic as the following:

$$h(t) = \frac{1}{n(t)} \sum_{i \in I(t)} [\mathbf{y}_i - \overline{\mathbf{y}}(t)]' V^{-1} [\mathbf{y}_i - \overline{\mathbf{y}}(t)] \tag{5}$$

where I(t), V, and $\overline{\mathbf{y}}(t)$ respectively represents a set of learning samples at node t, covariance matrix of entire learning samples, and sample mean vector of response variables at node t. This statistic has been originally used as multivariate monitoring statistic where response variables are continuous rather than categorical.

Let S(t) denote an index set of all possible splits at node t. Then the amount of homogeneity gain achieved by splitting node t is given by

$$\eta(t,s) = h(t) - \frac{n\{t_L(s)\}}{n} h\{t_L(s)\} - \frac{n\{t_R(s)\}}{n} h\{t_R(s)\} \tag{6}$$

Recall that the smaller h(t) the higher homogeneity. An optimum split for node t is therefore chosen by maximizing (6) and written by

$$s^* = \arg\max_{s \in S(t)} \eta(t,s)$$

3.2 Weight Sum of Node Homogeneities

As done in Siciliano and Mola[10], weight sum of Gini indices can be employed as a criterion for MRV-node splitting. This method has some advantages. Ease of computations will be one of them. This is because Gini indices are obtained for respective response variables. Relative importances between response variables are also accommodated in this criterion. However, a question arises as to how to obtain the importances. Therefore, for this approach to be operational, the answer to the question has to be prepared.

Gini index for response variable Y_g at node t is given by

$$h(g,t) = 1 - \sum_{j=1}^{K_j} \frac{n_j^2(g,t)}{n^2(t)} \tag{7}$$

where $n_j(g,t)$ represents the number of learning samples satisfying Y_g=j at node t. Letting w(g) denote the weight of response variable Y_g, we can obtain a weight sum of (7) as the following:

$$\begin{aligned} h(t) &= \sum_{g=1}^{M} w(g) h(g,t) \\ &= 1 - \sum_{g=1}^{M} \sum_{j=1}^{K_g} w(g) n_j^2(g,t) / n^2(t) \end{aligned} \tag{8}$$

where

$$\sum_{g=1}^{M} w(g) = 1 \text{ and } w(g) \geq 0 \ \ \forall g = 1,2,\cdots,M$$

Therefore, as explained in the previous section, we find an optimum split so that (6) can be maximized. Based upon the weight sum (8), Siciliano and Mola[10] proposed to use a predictability index instead of (6) which is given as

$$\tau(t,s) = \eta(t,s) / h(t)$$

It is noted that, unfortunately, there is no agreed procedures to determine weights. Although a data-analytic approach to finding an optimum weight set is sometimes applicable, they are in general given by other considerations, for example, management policy, priority in design and operation, analysis experiences, and cost structure.

3.3 Node Splitting by an Expected Loss

So far, we explained node splitting schemes of measuring node homogeneity. This section introduces an expected loss as a node splitting criterion. In this method, variable importance is used to find misclassification cost. Let $\hat{Y}_g$ denote a predicted class of gth response variable for a learning sample. Then a loss function can be defined by

$$L(\hat{Y}_g; Y_g) = \begin{cases} 1, & Y_g \neq \hat{Y}_g \\ 0, & Y_g = \hat{Y}_g \end{cases} \tag{9}$$

where g=1,2,…,M. Therefore, an MRV-loss function of M predictions can be rewritten by

$$L(\hat{\mathbf{y}}; \mathbf{y}) = \sum_{g=1}^{M} w(g) L(\hat{Y}_g; Y_g) \tag{10}$$

where $\mathbf{y}$ and $\hat{\mathbf{y}}$ are vectors of true classes and of predicted classes respectively. Taking mathematical expectation on (9), we can derive an expected loss for gth response variable as

$$L_g = \Pr(Y_g \neq \hat{Y}_g) = 1 - \Pr(Y_g = \hat{Y}_g) \tag{11}$$

Note that the expected loss (11) is itself a misclassification probability. Thus, taking expectation on (10) and substituting (11) again produces an MRV-expected loss which is written by

$$\begin{aligned} L &= \sum_{g=1}^{M} w(g) \Pr(Y_g \neq \hat{Y}_g) \\ &= 1 - \sum_{g=1}^{M} w(g) \Pr(Y_g = \hat{Y}_g) \end{aligned} \tag{12}$$

In practice, misclassification probability for gth response variable at a terminal node t can be estimated by

$$1 - \max_{1 \le j \le K_g} p_j(g,t) \tag{13}$$

where

$$p_j(g,t) = n_j(g,t) / n(t).$$

This estimation implies that all samples in node t are classified into a single class, the frequency of which is highest. Substituting (13) into (12), we can obtain an estimator of expected loss at node t as the following.

$$L(t) = 1 - \sum_{g=1}^{M} w(g) \max_{1 \le j \le K_g} p_j(g,t) \tag{14}$$

Therefore, total loss over the entire tree can be defined by

$$L(T) = \sum_{t \in T} p(t) L(t) \tag{15}$$

where T represents a set of terminal nodes in the tree. Also note that p(t)=n(t)/N where N is the total number of learning samples. The amount of loss reduction achieved by splitting node t is then given by

$$\lambda(t,s) = h(t) - \frac{n\{t_L(s)\}}{n} L\{t_L(s)\} - \frac{n\{t_R(s)\}}{n} L\{t_R(s)\} \tag{16}$$

3.4 Numerical Illustrations (I)

This section illustrates the node splitting procedure explained in Section 3.1 using a hypothetical data set which has two response variables. This data set includes 600 learning samples and the joint distribution of response variables is as shown in Table 2.

Table 2. Joint distribution of Y1 and Y2 at the root node

	Y2=1	Y2=2	sum
Y1=1	79	128	207
Y1=2	47	69	116
Y1=3	25	252	277
sum	151	449	600

Assume that there are two explanatory variables X_1 and X_2 with 3 categories respectively. Further assume that X_2 is ordinal categorical variable. To begin with, let us consider the following split condition; Take the left node if X_2=1, and take the right node elsewhere. Figure 3 illustrates a partial tree produced by this condition.

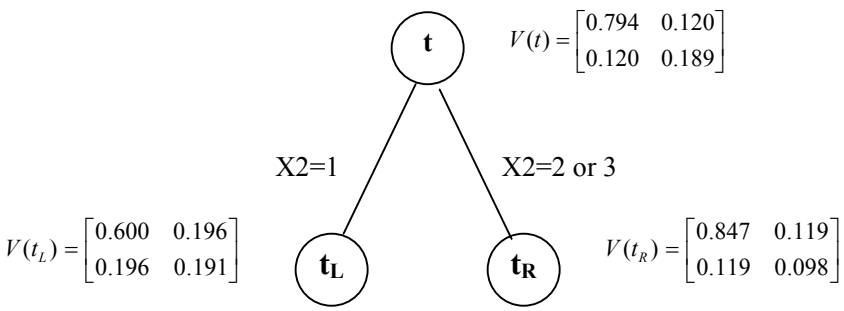

Figure 3. A node splitting with multiple response variables

In this figure V(t), V(t_L) and V(t_R) are sample covariance matrices of nodes t, t_L and t_R respectively. As a result of the node splitting, conditional joint distributions of Y_1 and Y_2 at nodes t_L and t_R are respectively given as the following tables.

Table 3. Joint distributions of Y1 and Y2 given X2=1(tL) and X2=2 or 3(tR)

(a) X2=1

	Y2=1	Y2=2	sum
Y1=1	28	0	28
Y1=2	47	0	47
Y1=3	25	34	59
sum	100	34	134

(b) X2=2 or 3

	Y2=1	Y2=2	sum
Y1=1	51	128	179
Y1=2	0	69	69
Y1=3	0	218	218
sum	51	415	466

Using (2), we can find multivariate entropy of node t from Table 2 as

$$h(t) = -\frac{79}{600}\log\frac{79}{600} - \frac{47}{600}\log\frac{47}{600} - \frac{25}{600}\log\frac{25}{600} - \frac{128}{600}\log\frac{128}{600} - \frac{69}{600}\log\frac{69}{600} - \frac{252}{600}\log\frac{252}{600} = 1.542$$

Similarly, from Table 3,

$$h(t_L) = 1.356 \text{ and } h(t_R) = 1.235$$

are obtained. Thus, using (6), entropy gain by splitting node t is given by

$$\eta(t) = 1.542 - \frac{134}{600} \times 1.356 - \frac{466}{600} \times 1.235 = 0.279$$

and this corresponds to improvement of 0.279/1.542=18.1%. Multivariate Gini index at node t is also calculated as

$$h(t) = 1 - (79^2 + 47^2 + 25^2 + 128^2 + 69^2 + 252^2)/600 = 0.740$$

by (3). Similarly, we can have

$$h(t_L) = 0.734 \text{ and } h(t_R) = 0.672$$

and thus the gain of Gini index η(t) is 0.054 or 7.3%. Table 4 shows entropy, Gini index, and their gains for 5 split conditions at node t. In this example, split 4 produces the best output irrespective of using either entropy or Gini index.

Table 4. Node splitting comparison of entropy and Gini index over s in S(t)

s	t_L	t_R	$p(t_L)$	$p(t_R)$	Entropy			Gini Index		
					$h(t_L)$	$h(t_R)$	η(t)	$h(t_L)$	$h(t_R)$	η(t)
1	X1=1	X1=2, 3	0.215	0.785	1.473	1.525	1.8%	0.740	0.728	1.3%
2	X1=2	X1=1, 3	0.182	0.818	1.280	1.576	1.3%	0.650	0.753	0.7%
3	X1=3	X1=1, 2	0.603	0.397	1.561	1.426	2.2%	0.741	0.714	1.2%
4	X2=1	X2=2, 3	0.223	0.777	1.356	1.235	18.1%	0.734	0.672	7.3%
5	X2=3	X2=1, 2	0.542	0.458	1.177	1.538	12.9%	0.658	0.743	5.8%

Other than entropy and Gini index, the covariance matrix determinant and Hotelling T^2 type statistic have been used in Zhang[7]. First, from Figure 3, $|V(t)|=0.136$, $|V(t_L)|=0.076$ and $|V(t_R)|=0.069$ are obtained. We can see that, under this criterion, homogeneity gain η(t) is 0.065 or 48.2%. Now, assuming that node t is the root node, we have

$$V^{-1} = \begin{bmatrix} 1.392 & -0.882 \\ -0.882 & 5.860 \end{bmatrix}$$

Thus, from (6), Hotelling T^2 type statistics of nodes t, t_L and t_R are calculated as

$$h(t) = \sum_{i \in I(t)} [\mathbf{y}_i - \bar{\mathbf{y}}(t)]' V^{-1} [\mathbf{y}_i - \bar{\mathbf{y}}(t)]/600 = 1.997$$

$$h(t_L) = \sum_{i \in I(t_L)} [\mathbf{y}_i - \bar{\mathbf{y}}(t_L)]' V^{-1} [\mathbf{y}_i - \bar{\mathbf{y}}(t_L)]/134 = 1.595$$

$$h(t_R) = \sum_{i \in I(t_R)} [\mathbf{y}_i - \bar{\mathbf{y}}(t_R)]' V^{-1} [\mathbf{y}_i - \bar{\mathbf{y}}(t_R)]/466 = 1.538$$

respectively. In this example, η(t)=0.446 or 22.3%.

3.5 Numerical Illustrations (II)

In this section node splitting methods described in Sections 3.2 and 3.3 are illustrated by another constructed example. First suppose that node t has 100 learning samples. Also suppose that there are two response variables and two explanatory variables. The following shows contingency tables between explanatory and response variables.

Table 5. Cross tabulations of X1, X2, Y1 and Y2 at node t

	Y1=1	Y1=2	Y1=3	sum
X1=1	38	15	7	60
X1=2	4	14	2	20
X1=3	8	1	11	20
sum	50	30	20	100

	Y1=1	Y1=2	Y1=3	sum
X2=1	30	0	0	30
X2=2	6	24	10	40
X2=3	14	6	10	30
sum	50	30	20	100

	Y2=1	Y2=2	Y2=3	sum
X1=1	15	40	5	60
X1=2	20	0	0	20
X1=3	5	0	15	20
sum	40	40	20	100

	Y2=1	Y2=2	Y2=3	sum
X2=1	19	10	1	30
X2=2	15	15	10	40
X2=3	6	15	9	30
sum	40	40	20	100

Using (6) yields Gini indices for Y_1 and Y_2 at node t as follows.

$$h(1,t) = 1-(0.5^2+0.3^2+0.2^2) = 0.62$$
$$h(2,t) = 1-(0.4^2+0.4^2+0.2^2) = 0.64$$

Thus the weighted Gini index (7) can be obtained by

$$\begin{aligned} h(t) &= w(1)h(1,t)+w(2)h(2,t) \\ &= 0.7\times 0.62+0.3\times 0.64 \\ &= 0.626 \end{aligned}$$

assuming the weights are given as w(1)=0.7 and w(2)=0.3. However, Siciliano and Mola[8] used node impurities to obtain weight for Y_i at node t which is written by

$$w(i,t) = h(i,t)/\sum_g h(g,t) \tag{17}$$

Using (17) produces two weights at node t as follows.

$$w(1,t) = 0.62/(0.62+0.64) = 0.49$$
$$w(2,t) = 0.64/(0.62+0.64) = 0.51$$

Thus weighted Gini index is obtained by

$$\begin{aligned} h(t) &= w(1)h(1,t)+w(2)h(2,t) \\ &= 0.49\times 0.62+0.51\times 0.64 \\ &= 0.630 \end{aligned}$$

The expected loss (14) is obtained as follows. First, from (13), misclassification probabilities for Y_1 and Y_2 are given by

$$L(1,t) = 1 - \max(0.5,0.3,0.2) = 0.5$$
$$L(2,t) = 1 - \max(0.4,0.4,0.2) = 0.6$$

respectively. Therefore, the expected loss (14) becomes

$$\begin{aligned} L(t) &= w(1)L(1,t) + w(2)L(2,t) \\ &= 0.7 \times 0.5 + 0.3 \times 0.6 \\ &= 0.530 \end{aligned}$$

Table 6 shows a comparison result of 6 scenarios for node splitting. Split 5 is best when weighted Gini index is used. On the contrary, for the expected loss, split 5 is best. Although not included in this table, split 2 is best under the weighting scheme proposed by Siciliano and Mola[8].

Table 6. Node splittings by weighted Gini index and by expected loss when (w1,w2)=(0.7,0.3)

s	t_L	t_R	$p(t_L)$	$p(t_R)$	Gini Index		Expected Loss		$\eta(t)$ %	$\lambda(t)$ %
					t_L	t_R	t_L	t_R		
1	X1=1,2	X1=3	0.8	0.2	0.573	0.487	0.483	0.390	11.3	25.9
2	X1=1,3	X1=2	0.8	0.2	0.593	0.322	0.448	0.210	14.0	36.1
3	X1=1	X1=2,3	0.6	0.4	0.512	0.605	0.357	0.550	12.3	30.7
4	X2=1,2	X2=3	0.7	0.3	0.602	0.628	0.494	0.523	2.6	19.6
5	X2=1,3	X2=2	0.6	0.4	0.485	0.585	0.362	0.468	16.1	35.5
6	X2=1	X2=2,3	0.3	0.7	0.146	0.653	0.110	0.571	20.0	30.8

By changing the weight, we can compare the above six scenarios again. These are depicted in the following figures. Although the two splitting criteria produce different results each other, it is hard to say which one is better or worse in this example. To answer the question, more rigorous comparisons are required in future. Nevertheless, one thing to underline is that weights for response variables have to be carefully chosen. This is because the best split can be altered by the weights. As shown by the figures, gains are smoothly increasing or decreasing along the weight. For example, split 3 yields the largest gain when $w(2) \geq 0.6$. But, in the lower range of w(2), split 6 is best for weighted Gini index(Figure 4a), and split 5 is best for expected loss(Figure 4b). We can see that, in particular, the preference of split 5 is sensitive to the choice of weights. On contrast, split 4 looks quite robust against w(2), which is because its gain of node splitting is too little.

4. Summary and Conclusions

Decision tree, as a machine learning approach, provides a promising way to building classification models from a large data set. Many applications are observed in the areas of biomedicine, public health, and business. Tree-based approaches would be also helpful for other industrial applications, for example like process monitoring and diagnosis. From a statistical viewpoint, ML techniques can be regarded as nonparametric statistics. In general, when knowledge about the population is insufficient, parametric methods have a difficulty in dealing with large complex data sets[4]. In such cases, machine learning techniques can be considered as a useful alternative. However, as the target population is more specific and domain knowledge increases, parametric statistics become more suitable to understanding the population. As stated in Bose and Mahapatra[1], decision trees are useful to deal with a large amount of data and have a high application capability. Nevertheless, in order to develop more reliable models, collaborative use with parametric methods such as regression and discriminant analysis should be pursued.

In many applications, we can observe that response variables are two or more. Medical diagnosis, customer credit prediction, and process monitoring are such examples. Mining classification rules based upon multiple response variables(MRV) is one of the most interesting problems in that MRV-data set can contain more information for explaining latent relationships. This article is concerned with constructing decision trees when there are two or more response variables in data set. Basically, decision trees in this article are binary classification trees.

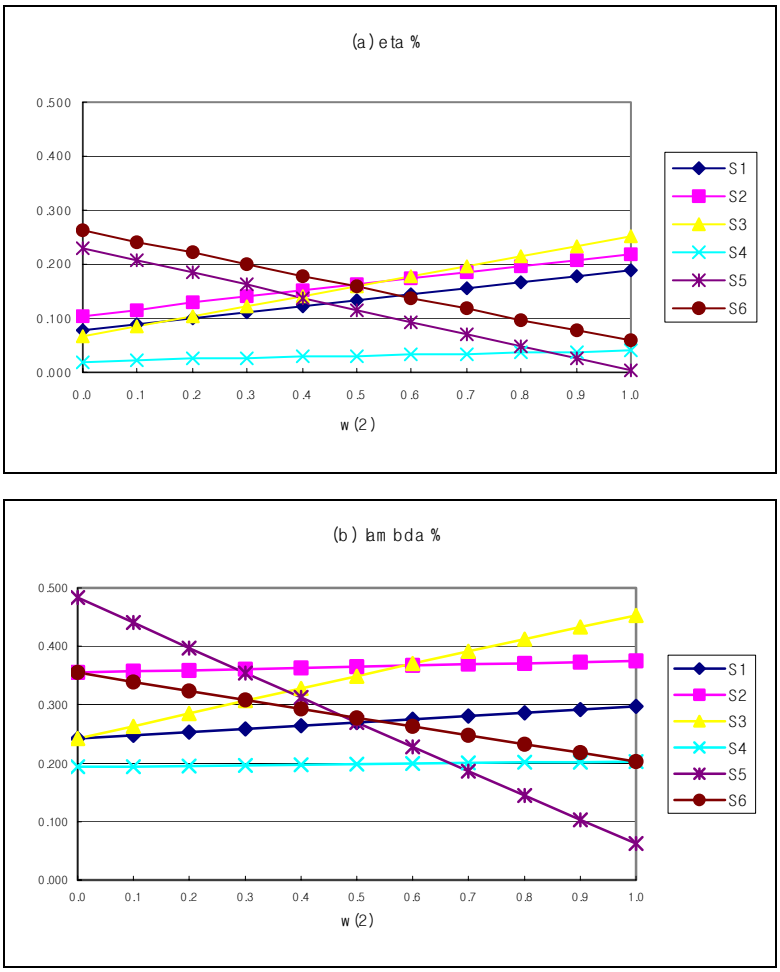

Figure 4. Gains of weighted Gini index and expected loss for six split scenarios

First we provided an overview on decision trees and investigated node splitting criteria with a single response variable(SRV). Then, we proposed three approaches to MRV-node splitting. The first approach employs entropy or Gini index to represent node impurity with MRV. Although it can be viewed as a natural extension to MRV situations, this approach has a limitation that individual characteristics between response variables are not accommodated. The second one is a weighting method: first Gini indices are obtained for each of response variables and then they are summed by predetermined weights. This approach definitely has advantages that computation load is relatively small and that variable importances are considered with ease. However, more studies are required as to how weights are determined as well as how the correlation structure between response variables is incorporated into the node splitting procedure. The third is also a weighting method. But it is different from the second one in that an expected loss is employed as a node splitting criterion instead of entropy or Gini index. By using this expected loss criterion, we can find a decision tree that minimizes misclassifications. Finally, illustrations of the three approaches are given by examples. However, for more rigorous comparison of the presented approaches, conducting extensive experimentations will be required. Even though this study is restricted to binary classification trees, our framework to deal with MRV situations could be still applied for other classification and regression trees. Future work in this area includes node splitting procedures in which response variables are numerical. Collaborating decision trees and statistical modeling methods would be also a fruitful subject to study further.

Acknowledgements: This work was supported by grant No. R05-2001-000-01406-0 from the Korea Science and Engineering Foundation. The authors also would like to acknowledge the support of Brain Korea 21 Project in 2002.

References

[1] Indranil Bose and Radha K. Mahapatra, Business Data Mining – A Machine Learning Perspective, Information & Management, Vol. 39, pp. 211-225, 2001.

[2] Jiawei Han and Micheline Kamber, Data Mining - Concepts and techniques, San Francisco, CA: Morgan Kaufmann, 2001.

[3] P. Langley and H. A. Simon, Applications of Machine Learning and Rule Induction, Communications of the ACM,

Vol. 38, No. 11, pp. 55-64.
[4] Katharina D. C. Stark and Dirk U Pfeiffer, The Application of Non-parametric Techniques to Solve Classification Problems in Complex Data Sets in Veterinary Epidemiology – An Example, Intelligent Data Analysis, Vol. 3, pp. 23-35, 1999.
[5] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone, Classification and Regression Trees, Boca Raton, FL: Chapman & Hall/CRC, 1998.
[6] Young M. Chae, Seung H. Ho, Kyoung W. Cho, Dong H. Lee and Sun H. Ji, Data Mining Approach to Policy Analysis in a Health Insurance Domain, International Journal of Medical Informatics, Vol. 62, pp. 103-111, 2001.
[7] Heping Zhang, Classification Trees with Multiple Binary Responses, Journal of the American Statistical Association, Vol. 93, No. 441, pp. 180-193, 1998.
[8] Roberta Siciliano and Francesco Mola, Multivariate Data Analysis and Modeling Through Classification and Regression Trees, Computational Statistics & Data Analysis, Vol. 32, pp. 285-301, 2000.
[9] Donald E. Brown and Clarence Louis Pittard, Han Park, Classification Trees with Optimal Multivariate Decision Nodes, Pattern Recognition Letters, Vol. 17, pp. 699-703, 1996.