Journal of the Korean
OR/ MS Society
Vol. 18, No. 2, August 1993

Splitting Algorithm Using Total Information Gain for a Market Segmentation Problem

Jae Kyeong Kim*, Chang Kwon Kim**, and Soung Hie Kim***

Abstract

One of the most difficult and time-consuming stages in the development of the knowledge-based system is a knowledge acquisition. A splitting algorithm is developed to infer a rule-tree which can be converted to a rule-typed knowledge. A market segmentation may be performed in order to establish market strategy suitable to each market segment. As the sales data of a product market is probabilistic and noisy, it becomes necessary to prune the rule-tree at an acceptable level while generating a rule-tree. A splitting algorithm is developed using the pruning measure based on a total amount of information gain and the measure of existing algorithms. A user can easily adjust the size of the resulting rule-tree according to his(her) preferences and problem domais. The algorithm is applied to a market segmentation problem of a medium-large computer market. The algorithm is illustrated step by step with a sales data of a computer market and is analyzed.

1. Introduction

Knowledge-based systems are currently an effective and popular method for providing computer-based decision support in certainly well circumscribed problem areas [2, 4, 21]. Knowledge representation in the most knowledge-based systems is usually based on a production system architecture composed of many condition-action production) rules[2]. One of the most difficult and time-consuming stages in the development of the knowlede-based system is a knowledge acquisition-the elicitation from the expert of a comprehensive and consistent set of rules. Recent works have been done on developing automatic methods, and one of these methods is that

* Dep. of Industrial Engineering, Taejun National University of Technology
** Software Development Division Consulting Team, Samsung Data Systems
*** Dep. of Management Information Systems, KAIST

of inducing rules from a set of examples or cases[8,11,14,15]. The use of inductive methods in knowledge acquisition is motivated by the hypothesis that an expert's decision processes can be inferred by studying the decisions he makes when presented with a set of representative examples or instances from the problem domain of interest. Examples can be provided by an expert or obtained from archival data. Each example has a number of attributes and can be classified into a particular (sub)class. The induction algorithm or splitting algorithm attempts to find a tree of rules(in the from of IF...THEN clauses) which will correctly classify all the examples on the basis of their attribute values. The CLS (concepts learning system) algorithm repatedly partitions the set of examples according to the attribute with the greatest discriminatory power [8]. A more improved version of CLS algorithm, ID3 (interactive dichotomizer 3) was developed by Quinlan [15] ID3 uses an information-theoretic measure to find the attribute, and continues the partitioning process until all of the examples in a subclass are not be partitoned. But the ID3 algorithm does not have a concept of statistical significance [14]. Hart [7] suggested using the chi-square contingency table test in place of the information measure (IM). Mingers[14] experimented with the use of the χ^2 statistic on noise data, and suggested using the G^2 statistic in the algorithm rather than χ^2 because G^2 is less sensitive to small frequencies. Raz [16] suggested an algorithm for grouping the values of qualitative classes while minimizing the loss of information about a dichotomous attribute. Also, he suggested a splitting algorithm based on Shannon's measure of mutual entropy[17]. Despite of many useful researches, it is fair to say that each approach is only the beginning to deal with many diverse and complicated problems. There is no one best way to rule induction problems and it is necessary to modify an existing algorithm to fit a specific domain. Recently, Carter and Catlett [3] compare a basic version of ID3 performance with an assessment method called credit scoring. Braun and Chandler [1] developed a commercially available software system, ACLS (analog concept learning system) which is used to analyze past examples and formulate decision rules, and to investigate one market prediction situation through the rule induction approach. Vanhonacker [20] discussed an information-theoretic approach testing the exact order of an individual's brand choice process and compared it with alternative tests that performing a similar task.

The market is conceptualized to be a population of customers with heterogeneous brand preferences[5]. Some competitiors will be in a better position to serve particular customer segments of the same market. A company, instead of competing everywhere, should identilfy the most attractive parts of the market that it could serve effictively. Thus companies are increasingly making use of target marketing. Target marketing helps identify market opportunities better in their marketing strategy. Companies can then develop suitable products for each target market. They can also adjust their prices, distribution channels, and advertising

to reach the target market efficiently[9]. To find the target market, it is necessary to carry out market segmentation considering competitive markets and products. A market segment is defined as a group of consumers who are homogeneous in terms of the probabilities of choosing the different brands in the product class. Although segmentation continues to be an important marketing concept, there is no one best way to segment markets, as each approach has merits and limitations depending on the product-market being considered and the managerial objectives for segmentation[5,22]. Therefore, it would be helpful to make a knowledge-based system which can classify a prespecified set of brands in a product class into suitable subclasses effectively. As a first step, it is necessary to develop a knowledge acquisition process suitable to a market segmentation problem. It is difficult and complicate to establish a marketing strategy for a large number of partitioned market segments. And it is also meaningless to establish a marketing strategy for very few market segments. In most cases, the terminal subclasses would have not the same objects in the resulting rule tree of a market segmentation problem. The subclasses would therefore become probabilistic. However, existing algorithms, if used without modifications, can not give a useful result to a market problem. Existing studies have used no stopping rules(for example ID3), or used an intermediate stopping rule which depends on only a local information (for example, χ^2 statisitc or G^2 statistic). So it is necessary to develop a splitting algorithm which can handle the probabilistic and noisy data and adjust the size of the rule-tree depending on problem domains and company's strategy.

The objective of this research is to develop a splitting algorithm suitable to a market segmentation problem. We suggest a stopping rule based on a global information measure, total information gain. And a STIG (Splitting using Total Information Gain) algorithm is developed using an existing rule (G^2 statistic) and a suggested stopping rule as a knowledge acquisition tool. The STIG algorithm is applied to segmenting a medium-large computer market in Korea, and is illustrated step by step. Although the STIG is developed as a market segmentation algorithm, it can be applied to other problem areas. And the STIG algorithm generates the same result as other existing splitting algorithms if it does not use the stopping rule to be suggested.

2. Splitting Algorithm for Knowledge Acquisition

Inductive inference attempts to discover production rules by analyzing a series of examples related to a particular problem after relevant attributes have been identified and their values have been determined for each instance [12]. Consider the set of examples depicted in Table 1.

It is sampled from the sales data of medium-large computer market in Korea. In brief, the aim is trying to predict a computer model from the attributes it possessed.

Table 1. An illustrative example set of computer market

Example No.	ATTRIBUTES						CLASS
	INDUSTRY	APPLICATION	SALES	PROFIT	EMPLOYEES	GROWTH	MODEL
1	EQUIP	MT	13516	25126	15720	50	HP1000-E
2	EQUIP	MT	1789	3314	7009	70	HP1000-E
3	EQUIP	MT	810	1394	4000	60	HP1000-E
4	STOCK	MG	738	2361	548	53	HP3000-68
5	FOOD	MG	2001	2487	5841	11	HP3000-68
6	CONST	SD	4101	11002	8525	7	IBM4331
7	BANK	SD	97	1453	195	12	IBM4331
8	STOCK	SD	688	-2246	356	-9	IBM4331
9	BANK	SD	77	1264	220	28	IBM4331
10	BANK	SD	100	2176	30	20	IBM-S60
11	CONST	MG	572	2326	500	17	IBM-S60
12	BANK	MG	160	1667	490	18	IBM-S60
13	EQUIP	MG	418	1074	1875	45	IBM-S60
14	BANK	MG	260	2246	76	20	IBM-S60

"SALES" unit : hundred thousand dollars
"PROFIT" unit : ten thousand dollars
"EMPLOYEES" unit : persons
"GROWTH" unit : percentage

MG : management
MT : management and development of technolgy
SD scientific development of technology

The last column indicates the class, ***Model***. The remaining column names identify the diagnostic attributes to be used in arriving at a classification decision. Each row of the table represents one example. The first example represents that a model HP1000-E was sold to a company of which industry was Equipment, application was MT(management), sales amount was 13516, profit was 25126, number of employees was 15720, and growth was 50%. In general, induction algorithms can be more appealing in terms of the time and effort involved if the final rule-tree is minimal. Minimal is used here in the sense that the procedures consider first those attributes about which there is the least uncertainty concerning their association with a particular class value. At any stage, the attribute selected is that attribute yielding the least entropy.

That is, the next attribute selected is that which has the least uncertainty associated with its occurrence with class values.

2.1 Information Theory

Of prime importance is the measure used to evaluate particular attributes, and this is based on Shannon's information theory. Shannon [12] proposed a qualitative measure of the amount of information which is strongly connected to the amount of uncertainty. In fact, the information is equal to the removed uncertainty. If an object of a class C can be classified into m different subclasses, $c_1,\ldots c_m$ and the probabiliy of an object being in class c_i is $p(c_i)$, then a self entropy $H(C)$ as the measure of the uncertaintty associated with a class C is:

$$H(C)=-\sum_c p(c_i)\log(p(c_i)).$$

The smallest value that $H(C)$ can take on is 0, corresponding to the case where C can take on a single value with probability 1. The upper limit of $H(C)$ occurs when all the values that C may take on are equally probable. The joint entropy of an attribute X and a class C is defined in a similar way :

$H(X,C) = -\sum_x \sum_c p(x,c)\log(p(x,c))$, with $p(x,c)$ being the joint probability.

The conditional entropy of C given an attribute X is defined as follows :

$$\begin{aligned} H(C|X) &= \sum_x p(x)\, H(C|X=x) \\ &= -\sum_x \sum_c p(x,c)\log(c|x)) \\ &= H(X,C)-H(X), \text{ where } H(C|X=x)= -\sum_c p(c|x)\log(p(c|x)). \end{aligned}$$

The conditional entropy denotes the uncertainity remaining in class C after the value of an attribute X is known. The fact, $H(C|X) \le H(C)$ indicates that the knowledge of the outcome of the X can only reduce the uncertainty of the C. The mutual entropy of X and C is defined as fo X and C is defined as follows :

$$\begin{aligned} H(X:C) &= H(X) + H(C) - H(X,C) \\ &= H(C) - H(C|X) \\ &= \sum_x\sum_c p(x,c)\log\left(p(x,c)/p(x)p(c)\right). \end{aligned}$$

The mutual entropy is a measure of the reduction in uncertainty due to the association or corre-

lation between X and C. If an attribute X totally determines C, the $H(C|X) = 0$, and $H(X:C) = H(C)$. The various mathematical properties of Shannon's entropy have been described at length elsewhere [6, 10]. The sample data in Table 1 has 14 examples, 2 nominal attributes, 4 numerical attributes and a calss. *Model* denotes a class and is expressed as C. Generally, splitting algorithms proceed by choosing an attribute X_i with permissible values $x_{i1}, \cdots, x_{im}$. The attribute *Application* is denoted as X_2, with permissible values x_{21}=MT, x_{22}= MG, and x_{23}=SD. Each example of the class(C) will have one of these values(x_{ij}) for X_i. This allows C to be split into subclasses $c_1, c_2, \cdots, c_m$, where c_1 contains those examples in C with value x_{i1} of X_i, c_2 contains those examples in C with value x_{i2} of X_i, and so on. To select the attribute about which there is the least uncertainty concerning their association with a particular class value, mutual entropy between each attribute and the class (C) is computed first. As the mutual entropy is a measure of the reduction in uncertainty, the attribute with maximum mutual entropy is considered first. To do this, it is conventional and convenient to represent the joint distribution of an attribute and a class in a contingency table. As an example, we calculated the mutual entropy between *Application* and *Model* using the contingency table illustrated in Table 2.

Table 2. Contingency table of MODEL and APPLICATION

MODEL	HP1000-E	HP3000-68	IBM 4331	IBM-S60	Total
APPLICATION = MT	3	0	0	0	3
APPLICATION = MG	0	2	0	4	6
APPLICATION = SD	0	0	4	1	5
Total	3	2	4	5	14

In Table 2, the set of examples of Table 1 is split on the value of *Application*. The self entropy of class, *Model* is calculated as follows:

$$H(Model) = -3/14 \log(3/14) - 2/14 \log(2/14) - 4/14 \log(4/14) - 5/14 \log(5/14) = 1.3337.$$

The conditional entropy of *Model* given *Application* is as follows:

$$H(Model|Application) = 3/14\ H(Model|Application=MT) + 6/14\ H(Model|Application=MG) + 5/14\ H(Model|Application=SD)$$

$$= 3/14(-3/3 \log(3/3) + 6/14 (-2/6 \log(2/6) - 4/6 \log(4/6)) + 5/14 (-4/5 \log(4/5) - 1/5 \log(1/5))$$

$$= 0.4515.$$

Thus, the mutual entropy of *Model* and *Application* is as follows:

H(*Model*:*Application*) = H(*Model*) − H(*Model*|*Application*) = 0.8822.

The mutual entropy of *Model* and other nominal attribute can be calculated in the same way. In the case of numerical attributes(integer-valued attributes), we not only have to calculate the mutually entropy but also have to compute the cut-point on which to split the attribute. Further expression can be found in the subsequent section.

2.2 Classification of Attributes

In the case of market segmentation problem, we classified attributes into three categories; dichotomous numerical attributes, dichotomous nominal attributes, and group attributes. Dichotomous numerical attributes and dichotomous nominal attributes have only two values and the value is determined by the splitting algorithm. Group attributes can have more than two values, and the value is determined by the user. For example, when a numerical attribute *Growth* in Table 3 has an integer value between −9 to 70, and if the user thinks that the values 20 and 45 are critical to subdivide the class set, the numerical attribute *Growth* is regarded as a group attribute. Similarly, the decision maker can use the nominal attribute *Industry* in Table 1 into a group attribute, by grouping it into four categories, e.g., Equipment, Bank, Stock, and the others. In this research, a nominal attribute with more than two values can be replaced with dichotomous nominal attributes by substituting for each attribute a set of dichotomous indicator variables that retain the original information [1,]. Consider the nominal attribute *Application* in Table 2 with three permissible values(MT, MG, and SD). The splitting algorithm substitutes *Application* with three dichotomous variables Appl−MT, Appl−MG, and Appl−SD which are defined as follows:

$$\text{Appl}-\text{MT}=\begin{cases}1, & \text{if } \textit{Application}=\text{MT}\\ 0, & \text{otherwise}\end{cases}$$

$$\text{Appl}-\text{MG}=\begin{cases}1, & \text{if } \textit{Application}=\text{MG}\\ 0, & \text{otherwise}\end{cases}$$

$$\text{Appl}-\text{SD}=\begin{cases}1, & \text{if } \textit{Application}=\text{SD}\\ 0, & \text{otherwise.}\end{cases}$$

It is developed a method substituting a numerical attribute into a dichotomous numerical attribute. This is done by finding a cut-point in which the numerical attribute has maximum discriminatory power. For example, to find a cut-point of a numerical attribute *Growth* in

Table 1, first, sort the class (***Model***) in increasing order of attribute *Growth* as Table 3.

Table 3. Sorted table of MODEL and GROWTH.

Rank	GROWTH	MODEL
1	−9	IBM4331
2	7	IBM4331
3	11	HP3000-68
4	12	IBM4331
5	17	IBM-S60
6	18	IBM-S60
7	20	IBM-S60
	20	IBM-S60
8	28	IBM4331
9	45	IBM-S60
10	50	HP1000-E
11	53	HP3000-68
12	60	HP1000-E
13	70	HP1000-E

Second, find the cut−point that has the maximum mutual entropy while cutting this attribute by ranking. The mutual entropy of *Model* and *Growth* when the cut−point is between 12 and 17 is calculated from the following Table 4.

Table 4. Contingency table of MODEL and GROWTH when cut-point is 12

MODEL	HP1000-E	HP3000-68	IBM 4331	IBM-S60	Total
GROWTH ≥ 12	0	1	3	0	4
GROWTH > 12	3	1	1	5	10
Total	3	2	4	5	14

$$H(Model|Growth)_{12-17}=4/14(-1/4\log(1/4)-3/4\log(3/4)) + 10/14(-3/10\log(3/10) -2\times1/10\log(1/10)-5/10\log(5/10)) = 0.9952.$$

$$H(Model|Growth)_{12-17}=H(Model)-H(Model|Growth)_{12-17}=1.3337-0.9952=0.3385.$$

Here, the subscript denotes the cut-point of *Growth*. Similarly, $H(Model:Growth)_{7-11}=0.2120$, $H(Model:Growth)_{21-28}=0.2445$, and so on. In this example, the cut-point with maximum discriminatory power is occurred between 45 and 46. So the class is divided into two subclasses; a subclass in which *Growth* is less or equal to 45 and a subclass in which *Growth* is more than 45. In generating a rule-tree, the class is continuously divided and the cut-point of a dichotomous numerical attribute is changed according to the varying subclasses.

3. Development of a Splittting Algorithm(STIG)

3.1 Notation

The notation to be used is as follows:

X_i : i^{th} attribute, $i = 1, \cdots, n$.

X : set of attributes, $X = \{ X_1, X_2, \cdots, X_n \}$

X_{ij} : j^{th} value of attribute X_i, $i = 1, \cdots, n$, $j=1, \cdots, m_i$.

C : initial set of all objects.

C^t : set of objects where the t^{th} splitting is occurred.

N : total number of objects in C.

n^t : number of objects in C^t.

$H(C^t)$: self entropy of C^t.

$H(C^t \mid X_i)$: conditional entropy of C^t given X_i.

$H(C^t : X_i)$: mutual entropy of C^t and X_i.

MT^t : maximum mutual entropy of C^t, i.e., $ME^t = \max_i H(C^t : X_i)$, $i=1, \cdots, n$.

TG^t : total amount of information gain after t^{th} splitting.

S^t : set of non-terminal subclasses after t^{th} splitting is performed.

The splitting algorithm may be characterized by an attribute selection rule and a stopping rule. The attribute selection rule specifies the attribute used for the next splitting step. Although it is hard to guarantee optimality in the sense that the set of attributes selected results in the largest amount of mutual information among all sets of the same size, the selection rule constitutes a reasonable heuristic. In this research, the attribute selection rule which

maximizes the mutual entropy is used so that it provides the largest amount of information about the (sub)class. The stopping rules of existing studies test each subclass whether more splits are possible, until only terminal subclasses remain [14, 17]. For each terminal subclass, the remaining attributes cannot provide any additional statistical significant information about the class. Although these stopping rules constitute reasonable heuristics, they may create many problems if used alone. As these stopping rules only consider whether the splitting of each subclass is statistically significant or not, it may produce a biased or bushy rule-tree. And as the data of market problems contain many nisiy or probabilistic data, it is necessary to control the size, the level, and the number of terminal subclasses of a resulting rule-tree. For such a purpose, it is developed a supplementary stopping rule which considers the total amount of information gain (or reduction of uncertainty) while generating a rule-tree. The next section explains the newly suggested stopping rule. For convenience, the existing stopping rule is named as a *stopping rule 1*, and the newly suggested stopping rule is named as a *stopping rule 2*.

3.2 Total Amount of Information Gain, a Measure for Pruning Process

To explain a *stopping rule 2*, let us first define some terminology.

Definition : Information Gain(IG)

When an attribute X^l is selected to split an arbitrary class C^l, IG^l is defined as a $H(C^l : X^l)$.

In this research, as X^l is selected under the condition that maximizes $H(C^l : X_i)$ $(i=1, \cdots, n)$, IG^l equals to ME^l. For example, a graphical representation of the rule-tree is provided in Fig. 1.

It is assumed that the initial class set is C^1. And a dichotomous nominal attribute X_3 is selected as a splitting attribute. So X_3 becomes X^1. Then C^1 is split two subclasses $C^1 \cap \{X_3 = x_{31}\}$ and $C^1 \cap \{X_3 = x_{32}\}$. Further $C^1 \cap \{X_3 = x_{32}\}$ is assumed to be split. Then $C^1 \cap \{X_3 = x_{32}\}$ becomes C^2. As a group attribute X_5 is selected, C^2 is split into three subclasses $C^2 \cap \{X_5 = x_{51}\}$, $C^2 \cap \{X_5 = x_{52}\}$ and $C^2 \cap \{X_5 = x_{53}\}$ according to the value of X_5. The dashed nodes such as a $C^2 \cap \{X_5 = x_{51}\}$ represent terminal subclasses, which means additional splitting is no longer statistically significant. The number of objects in C^l is denoted as n^l. In the example rule-tree of Figure 1, *infromation gain* of C^1 is the mutual entropy of C^1 and X_3, i.e. $IG^1=ME^1=H(C^1 : X_3)$. Similarly, $IG^2=ME^2=H(C^2 : X_5)$. Now we consider the concept of *Total Amount of Information Gain*(TG). TG is defined as follows:

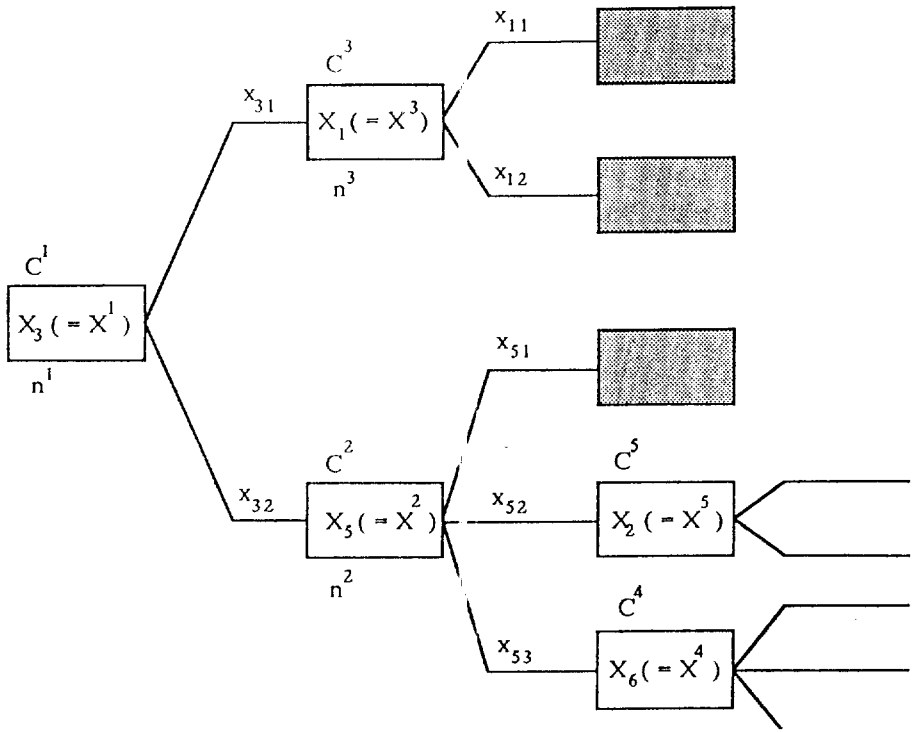

Fig. 1. An illustative rule-tree

Definition : Total Amount of Information Gain('TG)

When t^{th} splitting is occurred at C^t and the number of objects of C^t is n^t, TG^t is defined as $TG^t=TG^{t-1}+(n^t/N)IG^t$, and $TG^0=0$.

TG^t is a summation of information gain from initial state to the state after the t^{th} splitting is occurred. The difference between TG^t and TG^{t-1} is the reduction in uncertainty due to the splitting of subclass C^t. The properties related to TG is repreaented as follows:

Property : 1. TG^t is monotonically increasing.

2. The smallest value that TG^t can take on is 0.

3. The largest value that TG^t can take on is H(C), where C is an initial class.

Proof. As N, n^t and IG^t (ME^t) are greater than or equal to zero, $TG^t \geq TG^{t-1}$. So, TG^t is monotonically increasing. Next, the smallest value of TG^t occurs in the case where an initial class (C) is not split at all. In such a case, the value of TG^t is zero. The upper limit of TG^t occurs when all the terminal subclasses have only one kind of object(s). Assume an initial

selection attribute X^1 totally determines initial class C. In this case, $H(C \mid X^1)$ becomes zero. So, $TG^1=IG^1=ME^1=H(X^1:C)=H(C)-H(C \mid X^1)=H(C)$. ■

Using the properties of TG, we suggest a measure $TG^1/H(C)$ which is monotonically increasing and has a value between 0 and 1. Now we make a stopping rule 2 based on $TG^1/H(C)$. The algorithm will be stopped when $TG^1/H(C)$ exceeds for the first time a given value, say 65%, 85% or 95%. This implies that the user willingly submit 35%, 15% or 5% of the uncertainty related the class.

3.3 A Splitting Algorithm Using Total Amount of Information Gain, STIG

A splitting algorithm, STIG(Splitting using Total Information Gain) algorithm is developed in this research. The STIG uses a stopping rule 1 and a rule 2. The G^2 statistic is used for stopping rule 1. The distribution of the G^2 statistic is reported to be a very close approximation to the χ^2 distribution, and the value of the G^2 in a given class is equal to ME multiplied by 2N, where N is the number of examples of a given class [19]. And in the tails of the distribution, the G^2 statistic can be a better approximation than the χ^2 statistic. In STIG, G^2 is used to select the attribute with a maximum discriminatory power. And a stopping rule 1 using G^2 statistic judges the significance whether to split or prune the tree based on the information of subcalss. If the value of $TG/H(C)$ becomes larger than a predefined threshold, splitting is not occurred any more. The reason for using the stopping rule 2 for the pruning process is to prevent the possibility of generating a bushy rule-tree in noisy or probabilistic data, and to balance the overall structure of a rule-tree. Therefore, a stopping rule 1 and 2 are regarded as a local and a global stopping rule, respectively. Fig. 2 represents an overall flow of STIG.

In step(14), choosing one subclass among non-terminal set(S^t) is performed as follows: Suppose the member of S^t is composed of c_1, c_2, ···, c_s. And the number of objects of the non-terminal subclass c_k is assumed to n_k. Then, the subclass is selected as C^{t+1} which maximizes$(n_k/N)ME_k$, where ME_k is maximum mutual entropy of c_k(k=1, 2, ···, s .

The subclass which has more objects and can give more information gain (ME) is chosen at next splitting subclass. Thus, such a subclass is expected to reduce the more uncertainty of the class. In the STIG algorithm, the value of α_1 and α_2 are given subjectively by user. Here, α_1 value is a significance level and α_2 value is a threshold which is the minimum necessary amount of information in the problem domain. The values of α_1 and α_2 determine the degree of pruning a rule-tree via a

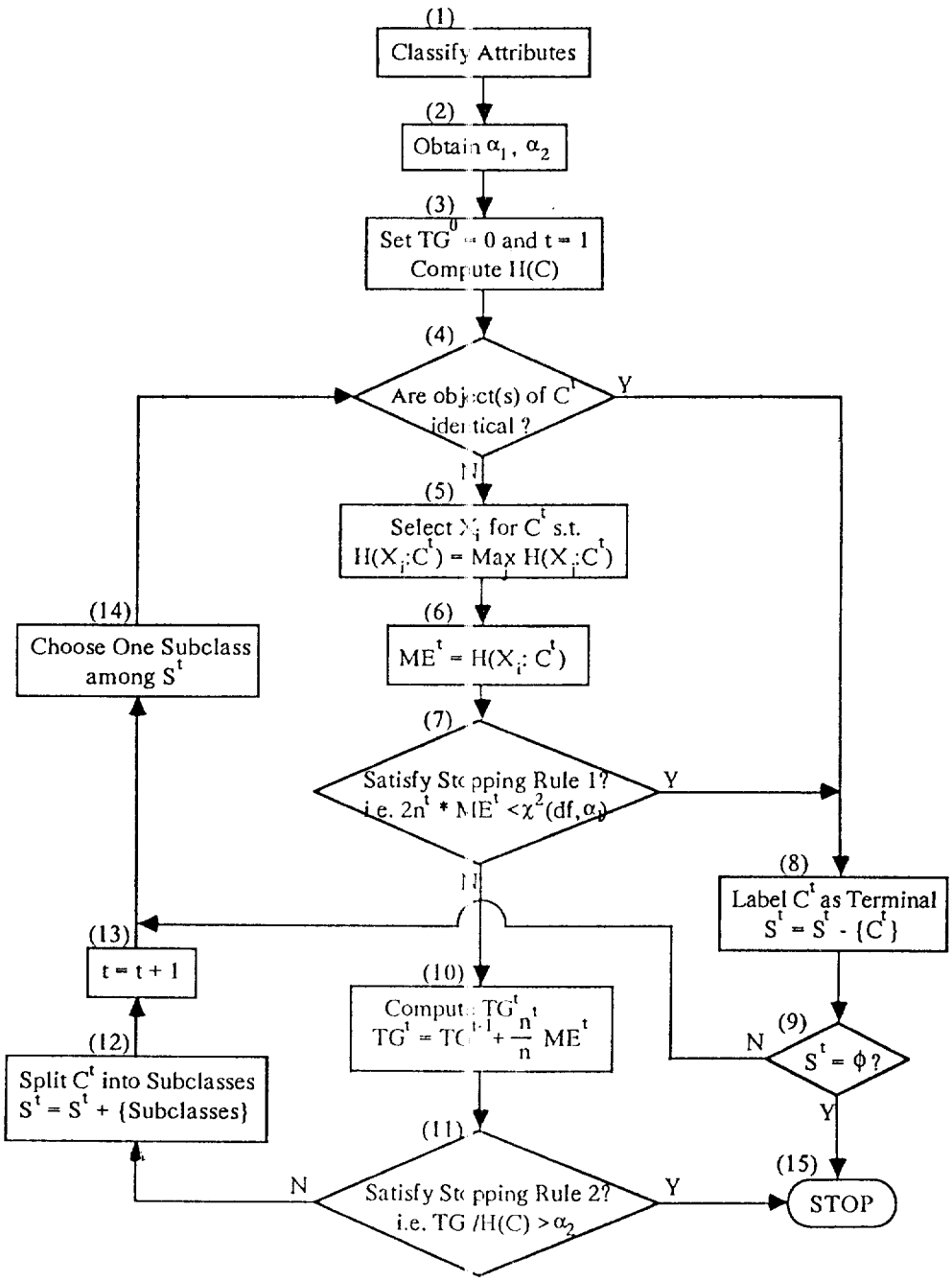

Fig. 2. The overall flow of STIG

stopping rule 1 and rule 2, respectively. The higher the values of α_1 and α_2, the more the number of subclasses increase. Segmenting process of medium-large computer market using STIG algorithm is illustrated in next section. And STIG algorithm is compared and analyzed in various values of α_1 and α_2.

4. Application of STIG to a Market Segmentation Problem

To illustrate a STIG algorithm applied to a market segmentation problem, we provide a data of a medium-large computer market in Korea in 1989. There are twelve companies in a medium-large computer market in Korea, but a sample set of examples in Table 1 are used to demonstrate the STIG algorithm step by step. The nominal attribute, *Industry* is designed to have 16 values (BANK, FOOD, EQUIPMENT, etc.), and *Application* has three values, MG, MT, and SD. The numerical sttributes, *Sales*, *Profit*, *Employees*, and *Growth* have integer values. The sequences of STIG algorithm of a set of examples of Table 1 can be summarized as follows:

ITERATION 0

Step 1, first, a decision maker classifies attributes. It is assumed that the attributes are classified as follows:

dichotomous nominal attribute−*Industry*(X_1),

dichotomous numerical attributes−*Sales*(X_3), *Profit*(X_4), *Employees*(X_5), *Growht*(X_6),

group attribute−*Application*(X_2) (1=MT, 2=MG, and 3=SD).

Step 2, the values of a_1, a_2 are obtained from the decision maker. In this, it is assumed that a_1=0.025, and a_2=0.85.

Step 3, set TG^0=0 and t=1, H(G)=1.33374, an initial class set(examples from #1 to #14) is termed as C^1, and n^1=N=14.

ITERATION 1

Step 4, as the objects in C^1 are not identical, go to 5.

Step 5, for each X_i (i=1, ···, 6), mutual entropy $H(X_i : C^1)$ is calculated. X_2, *Application* is selected because X_2 and C^1 produces maximum mutual entropy.

Step 6, ME^1=H(*Application* : C^1)=0.8822.

Step 7, as $\chi^2(6, 0.025)=14.4494$, $2n^1\times ME^1=2\times14\times0.8822=24.7024>\chi^2(6, 0.025)$, C^1 does not satisfy stopping rule 1. So, go to step 10.

Step 10, $TG^1 = ME^1 = 0.8822$.

Step 11, as $H(C^1)=1.33374$, $TG^1/H(C^1)=0.6614 < \alpha_2$, C^1 does not satisfy Stopping rule 2. Go to step 12.

Step 12, C^1 is split into three subclasses, $C^1 \cap \{ X_2 = x_{21} \}$, $C^1 \cap \{ X_2 = x_{22} \}$ and $C^1 \cap \{ X_2 = x_{23} \}$,

$C^1 \cap \{ X_2 = x_{21} \} = \{ \#1-\#3 \}$, $C^2 \cap \{ X_2 = x_{22} \} = \{ \#4, \#5, \#11-\#14 \}$ and

$C^3 \cap \{ X_2 = x_{23} \} = \{ \#6-\#10 \}$, $S^1 = \{ C^1 \cap \{ X_2 = x_{21} \}, C^1 \cap \{ X_2 = x_{22} \}, C^1 \cap \{ X_2 = x_{23} \} \}$.

ITERATION 2

Step 13, t=2.

Step 14, choose a subclass $C^2 = C^2 \cap \{ X_2 = x_{23} \}$ and $S^2 = \{ C^1 \cap \{ X_2 = x_{21} \}, C^1 \cap \{ X_2 = x_{23} \} \}$.

Step 4, as objects of C^2 are not identical, go to 5.

Step 5, mutual entropy $H(X_i : C^2)$ is calculated for each attribute. X_3, *Sales* is selected.

Step 6, $ME^2 = H(Sales : C^2)=0.6365$.

Step 7, as $\chi^2(1, 0.025)=5.0239$, $2n^2\times ME^2=2\times6> 0.6365=7.6382 > \chi^2(1, 0.025)$, C^2 does not satisfy a stopping rule 1. So go to step 10.

Step 10, $TG^2 = TG^1+(n^2/N)\times ME^2=0.8822+6/14\times0.6365=1.1550$.

Step 11, $TG^2/H(C)=0.8660 > \alpha_2$, go to 15.

Step 15, Stop.

A resulting rule-tree of a set of examples of Table 1 is displayed in Fig. 3.

```
Attribute-Name, TG, TG/H(C), ME, CHI-Value, Num_of_Exam, G-Value

APPLICATION  0.8822, 0.6614, 0.8822, 14.4457, 14, 24.7024
+--APPLICATION = MT
|          HP1000-E     3
+--APPLICATION = MG
|       SALES  1.1550, 0.8660, 0.6365,  5.0247,  6,  7.6382
|       +--SALES <= 572
|       |           IBM-S60     4
|       +--SALES >  572
|                  HP3000-68     2
+--APPLICATION = SD
           IBM-S60     1
           IBM4331     4
```

Fig. 3. The rule-tree of Table 1(α_1=0.025, α_2=0.85)

From this rule-tree, a set of production rules are derived as follows:

Rule 1 : IF Application is Management & Technology
THEN HP1000-E(3/3)
Rule 2 : IF Application is Management
AND Sales < =572
THEN IBM-S60(4/4)
Rule 3 : IF Application is Management
AND Sales > 572
THEN HP3000-68(2/2)
Rule 4 : IF Application is Scientific Technology
THEN IBM-S60(1/5)
IBM4331(4/5)

The number in the parenthesis of the THEN part indicates the portion of that brand. In most cases, it is natural that a group of companies under a similar conditions choose the different brands in the product class. Table 5 represents an example set of products of Hewlett Packard and IBM sampled from 12 companies of medium-large computer market in Korea.

Table 5. Example set of products HP and IBM.

Company	MODEL	Number	Company	MODEL	Number
HEWLETT PACKARD	HP3000-48	6	IBM	IBM-S60	9
	HP3000-68	9		IBM3083	7
	HP9000	13		IBM4331	6
				IBM4361	8

A rule-tree generated from the data of Table 5 is illustrated in Fig. 4. The Fig. 4. indicates that the market segments are mostly heterogeneous and the data are probabilistic and noisy.

The size and level of a rule-tree are varied according to various values of α_1 and α_2. Fig. 5 illustrates the relation between α_2 value and the number of subclasses when $\alpha_1 = 0.05$ and $\alpha_2 = 0.10$, respectively. For illustration the relation, 19 examples are sampled from the sales data of a medium-large computer market.

As known from Fig. 2, the α_1 and α_2 values determine the degree of pruning a rule-tree via a stopping rule 1 and rule 2, respectively. The higher the values of α_1 and α_2, the more the number of subclasses increase. This is obvious that the higher values of α_1 and α_2 lessens the extent

```
Attribute-Name, TG, TG/H(C), ME, CH -Value, Num_of_Exam, G-Value

SALES  0.3698, 0.1935, 0.3698, 14.4 94, 58, 42.9016
+--SALES <= 688
|      PROFIT  0.8594, 0.4497, 0.290 , 11.1433, 25, 14.4976
|      +--PROFIT <= 544
|      |  INDUSTRY  0.9672, 0.5060,  .5210, 9.3484, 12, 12.5038
|      |  +--INDUSTRY = EQUIP
|      |      |            HP9000    3
|      |      |           IBM-S60    1
|      |      +--INDUSTRY = NOT EQUIP
|      |             EMPLOYEES  1.2164  0.6364, 0.5623,  7.3778,  8,  8.9974
|      |             +--EMPLOYEES <= 3 6
|      |             |           IBM-S6     1
|      |             |           IBM433     5
|      |             +--EMPLOYEES >  3 6
|      |                        HP3000-4     2
|      +--PROFIT >  544
|            GROWTH  1.1388, 0.5958, 0.3584,  7.3778, 13,  9.3182
|            +--GROWTH <= 3
|            |            HP9000    3
|            |           IBM4361    1
|            +--GROWTH >  3
|                  GROWTH  1.2656, 0 6622, 0.3175,  5.0239,  9,  5.7156
|                  +--GROWTH <= 9
|                  |           IBM-S6     6
|                  +--GROWTH >  9
|                              HP900     2
|                             IBM-S6     1
+--SALES >  688
       GROWTH  0.5784, 0.3026, 0.366 , 12.8325, 33, 24.1941
       +--GROWTH <= 29
       |      EMPLOYEES  0.7344, 0.38 2, 0.5324, 12.8325, 17, 18.1014
       |      +--EMPLOYEES <= 5841
       |      |      EMPLOYEES  1.0584  0.5538, 0.6616,  9.3484,  8, 10.5850
       |      |      +--EMPLOYEES <= 1 00
       |      |      |           IBM436     5
       |      |      +--EMPLOYEES >  1 00
       |      |                HP3000-4     1
       |      |                HP3000-6     1
       |      |                   HP900     1
       |      +--EMPLOYEES >  5841
       |                    IBM3083    7
       |                    IBM4331    1
       |                    IBM4361    1
       +--GROWTH >  29
                 HP3000-68    5
                 HP3000-48    3
                 HP3000-68    3
                    HP9000    4
                   IBM4361    1
```

Fig. 4. The rule-tree of products HP and IBM(α_1=0.025, α_2=0.65)

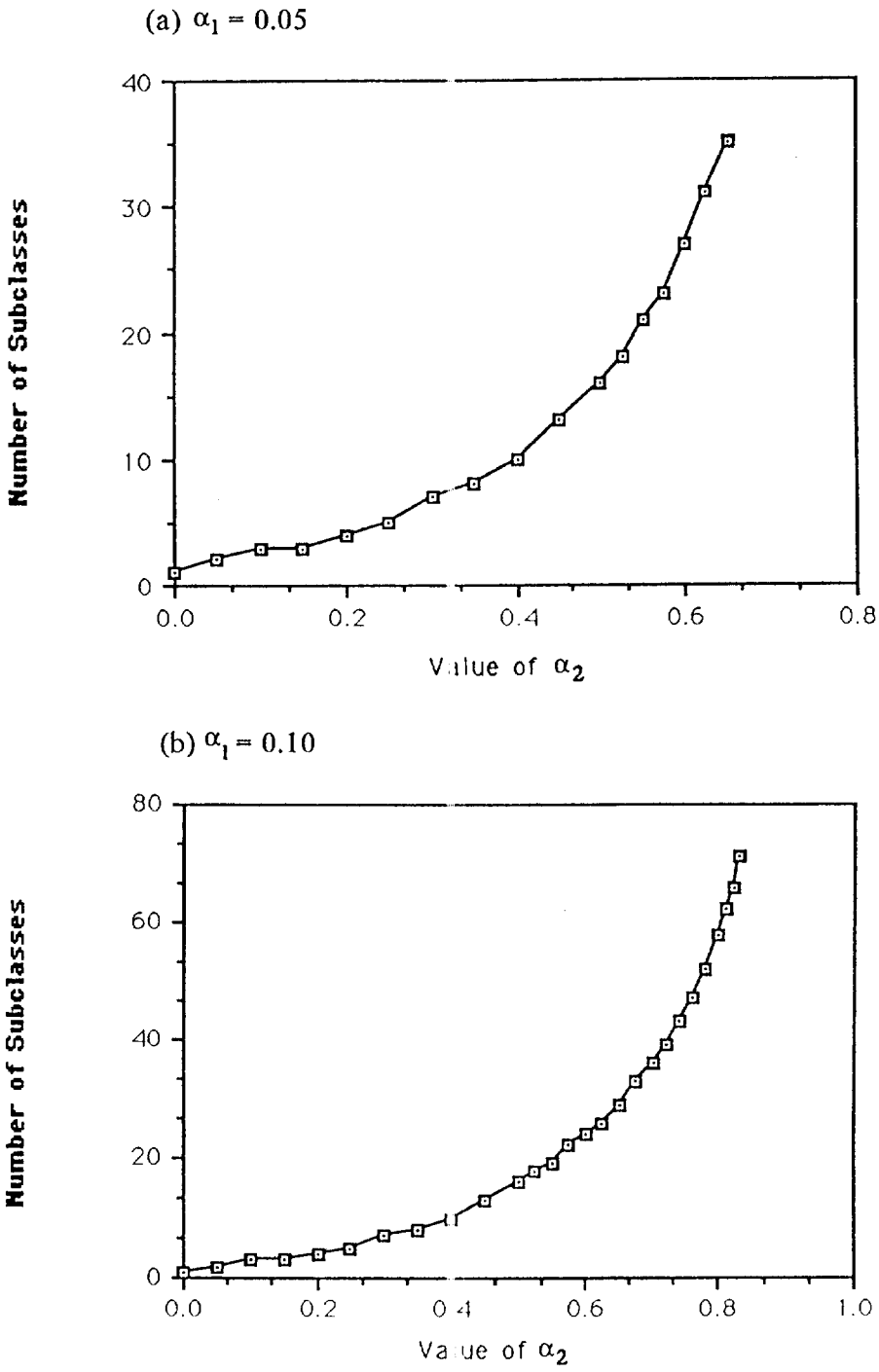

Fig. 5. Relation between α_2 and number of subclasses.

of pruning a rule-tree from a step (7) and step (11) of Fig. 2. And in the case $\alpha_2=1$, the STIG algorithm generates an identical rule-tree of existing studies[13, 16] because stopping rule 2 can not prune the tree. The user can adjust the size of a rule-tree, number of objects in subclasses, and significance of the result by deciding appropriate values of α_1 and α_2 according to his(her) preferences and problem domains. The STIG algorithm has been applied to make a rule-tree of the medium-large computer market in Korea for Samsung Hewlett-Packared Company. The STIG algorithm decreases the amount of time and effort for anlayzing market segments and establishing market strategy, and it is believed to increase the acceptance of a splitting algorithm as a useful tool.

5. Conclusions

This paper has introduced a methodology of inducing production rules from the set of examples in computer market in order to automate the development of knowledge bases is expert systems. The splitting algorithm, STIG is developed and applied to the market segmentation of the medium-large computer market in Korea. The rule-tree derived by a STIG algorithm has been used to build a knowledge-base This knowledge-base can help to assign competitive products to appropriate market segments and make an appropriate strategy at each market segment.

The STIG algorithm can be characterized by the attribute selection rule and two types of stopping rule for the pruning process. In splitting a (sub)class, STIG selects the attribute which provides the largest amount of information about the(sub) class and the attribute.

Two types of stopping rule are used in STIG to determine whether additional splitting of a subclass is justified or not. Generally, existing induction methods use a ***stopping rule 1***, which determines additional splitting using the information (G^2-statistic) of a given subclass. In contrast, a ***stopping rule 2*** suggested in this paper considers the total amount of information gain for splitting the subclasses. A ***stopping rule 2*** prevents a rule-tree to be too sensitive in small frequencies, and builds the rule-tree in the direction of reducing the entire uncertainty. The user can adjust the size of rule-tree, number of objects in subclasses, and significance of the result by deciding appropriate values of α_1 and α_2.

However, the knowledge-base resulted from the sales data can not offer an exact behavior of customers. The knowledge of the specification of each product(computer model) is thought to be also appended to a generated knowledge-base. Though the STIG algorithm is developed for

a market segmentation problem, it can be also used in other problem areas. They would be a promising further research area to apply the STIG algorithm to other areas and to develop a knowledge-based system relating the STIG algorithm.

References

[1] Braun, H. and J.S. Chandler, "Predicting stock market behavior through rule induction: An application of the learning from example approach," *Decision Science*, Vol.18 (1987), pp.415-429.

[2] Buchanan, B.G. and E.H. Shortliffe, *Rule-Based Expert System: The MYCIM Experiments of the Stanford Heuristic Programming Project*, Addison Wesley, 1984.

[3] Carter, C. and J. Catlett, "Assessing credit card applications using machine learning," *IEEE Expert*, Vol. 1(1987), pp.71-79.

[4] Clancy, W.J. and B.G. Buchanan, *Readings in Medical Artificical Intelligence: The First Decade*, Addison Wesley, 1984.

[5] Grover, R. and V. Srinivasan, "A simultaneous approach to market segmentation and market structuring," *J. of Marketing Research*, vol 24 (1987), pp.139-153.

[6] Guiase, S., *Information Theory with Applications*, McGraw-Hill New York, 1977.

[7] Hart, A., "Experience in the use of an induction system in knowledge engineering," in *Research and Developments in Expert System* (M. Bramer, Ed.), Cambridge University Press, 1984.

[8] Hunt, E.B. J. Martin and P.T. Stone, *Experiments in Induction*, Academy Press, New York, 1966.

[9] Kolter, P., *Marketing Management-Analysis, Planning, and Control, 5th ED.* Prentice Hall, New Jersey, 1984.

[10] Mathi, A.M. and P.N. Rathie, *Basic Concepts in information Theory and Statistics-Axiomatic Foundations and Applications*, John Wiley & Sons, New York, 1975.

[11] Messier, W.F. and J.V. Hansen, "Inducing rules for expert system development: an example using default and bankruptcy data" *Management Science* Vol.34 (1988), pp. 1403-1415

[12] Michalski, R.S., "A theory and Methodology of Inductive Learning," In *Machine Learning. An Artificial Intelligence Approach* (R S. Michalski *et al.*, Eds). pp.83-134. Palo Alto, California, 1983.

[13] Mingers, J., "Expert systems-experiments with rule induction," *J. Opl Res. Soc.*, Vol.37 (1986), pp.1031-1037.

[14] Mingers, J., "Expert systems-rule induction with statistical data," *J Opl Res. Soc.*, (1987), pp.39-47.

[15] Quinlan, J.R., "Learning efficient classification procedures and their application to chess end games," In *Machine Learning: An Artificial Intelligence Approach* (Michalski, R.S., *et al.*, Eds). pp.463-482. Palo Alto, California, 1983.

[16] Raz, T. and J. Goldman, "A grouping algorithm for qualitative data with a dichotomous outcome variable," *IIE Transactions*, Vol.17 (1985), pp.168-174.

[17] Raz, T., "A splitting algorithm for identifying qualitative variables related to a dichotomous dependent variable," *IIE Transactions*, Vol. 19 (1987), pp.190-198.

[18] Shannon, C.E., "A mathematical theory of communication," *Bell Syst. Tech. J.*, Vol.27 (1948), pp.379-423.

[19] Sokal, R. and F. Rohlf, *Biometry*, Freeman San Francisco, 1981.

[20] Vanhonacker, W.R., "Testing the exact order of an individual's choice process in an information theoretic framework," *J. of Marketing Research*, Vol.22 (1985), pp.337-387.

[21] Waterman, D.A., *A Guide to Expert Systems*, Addison Wesley, 1986.

[22] Wind, Y., "Issues and advances in segmentation research," *J. of Marketing Research*, Vol.15(1978), pp.317-337.