TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

1. INTRODUCTION

Knowledge discovery is the non-trivial extraction of implicit, previously unknown and potentially useful information from data [1]. This information may not be readily obvious in a large body of data due to the volume of overly specific data. DB-Discover is a machine learning program that uses attribute-oriented generalization to discover new and non-trivial information that is implicit in a database [7].

Attribute-oriented generalization is accomplished by repeatedly replacing specific attribute values in a relation with more general concepts. The generalization is performed on an attribute by attribute basis. As related specific attribute values are grouped together into more general concepts, some tuples in the relation become redundant. By eliminating all but one of the redundant tuples, the total number of tuples in the relation is reduced. The generalization continues until a specified maximum number of tuples is obtained.

Each attribute has an associated concept hierarchy, defined by a domain expert, that guides the generalization of that attribute. A *concept hierarchy* is represented as a tree structure that has some representation of all possible attribute values as leaves and a single most general concept called ANY at the root. The interior nodes of the tree represent increasing levels of generalization as the tree is ascended from the leaves to the root.

The degree of generalization is governed by two thresholds. An *attribute threshold* T_a for each attribute is used to specify the maximum number of distinct attribute values for that attribute that are permitted in the final generalized relation. The *table threshold* T_t specifies the number of tuples permitted in the final generalized relation. The generalization process is done by first reducing the number of distinct values for all attributes being considered to less than or equal to their attribute thresholds. The resulting relation is called the *prime relation*. If the prime relation contains more tuples than the table threshold, an attribute is chosen by some method for further generalization. This process continues until the number of tuples in the relation is less than or equal to the table threshold. The resulting relation is called the *final generalized relation*.

Currently, DB-Discover chooses the attribute in the prime relation that has the most distinct values for further generalization toward the final generalized relation. This paper outlines an investigation of different strategies that could be used for selecting the attributes for this generalization. The comparison criteria used reflect the efficiency of the strategies for converting the prime relation to the final generalized relation and the ability of the strategies to produce new and interesting results in that relation.

The criteria used to compare the different strategies are discussed in Section 2. Section 3 provides a discussion of the strategies examined, examples to illustrate their behavior and a time complexity analysis of the strategies. Section 4 provides a brief outline of the structure of DB-Discover. Section 5 details the experimental program. Section 6 shows the results of the experimental program. Section 7 provides conclusions and recommendations.

2. CRITERIA FOR THE EVALUATION OF SELECTION STRATEGIES

To evaluate the strategies for selecting an attribute for generalization from the prime relation toward the final generalized relation, criteria for comparison must be established. The purpose of the DB-Discovery program is to perform the efficient extraction of useful, interesting and implicit information from a large body of data stored in a database by generalizing the data to an understandable level. While the efficiency of the strategy for moving from the prime relation to the final general relation can be easily evaluated, the program's ability to extract useful, interesting information and provide that information at an understandable level of detail is more difficult to measure and evaluate.

Hamilton and Fudger [6] proposed two measures of interestingness to quantify the significance of information discovered from databases using the DBLEARN machine learning program. The first measure, referred to as I_1, is the number of attribute values in the final generalized relation that correspond to non-leaf, non-ANY concepts in the appropriate concept hierarchy. Occurrences of the same attribute value are counted separately. The original values of the attributes and the ANY node of a concept tree do not provide any new information that cannot obtained by direct database queries. The set

of non-ANY, non-leaf attribute values in output relation R is represented by V_R and defined as

$$V_R = \{(t,a,R_{t,a}) \mid t \in R,\ a \in A,\ R_{t,a} \neq ANY \text{ and } R_{t,a} \text{ is not a leaf}\}$$

where R is the relation being analyzed, t is a tuple in relation R, a is an attribute in the set of attributes A and $R_{t,a}$ is the value a in the tuple t of R. The measure of interestingness I_1 is then expressed by

$$I_1(R) = | V_R |$$

The second measure of interestingness, referred to as I_2, considers the depths and the weighted heights of concepts in the appropriate concept hierarchy. A concept that is farther from the ANY concept is more likely to provide specific information. A concept that is farther from the leaves of a concept tree is conceptually farther from the base values for the attribute.

The *depth* of a node in a concept hierarchy tree is defined so that the depth of the root node is 0 and the depth of any other node is 1 more than the depth of its parent. The *weighted height* of a node is a function of the number of leaf nodes of the subtree that has that node as the root and a weight value, which is the sum of the distances from the root node of the subtree to the leaf nodes of the subtree.

The number of leaf nodes of a subtree in a concept hierarchy that has node t as its root is determined using the equations

$$n(t) = 1 \qquad \text{if } t \text{ is a leaf node}$$

$$n(t) = \sum_{c \in C(t)} n(c) \qquad \text{otherwise}$$

where $n(t)$ is the number of leaf nodes, $C(t)$ is the set of child nodes of the node t and c is a member of that set. The *weight* of the node t is determined using the equation

$$w(t) = n(t) + \sum_{c \in C(t)} w(c)$$

The *weighted height* of the node t is then determined using the equation

$$wh_t = w(t)/n(t)$$

The interestingness of the attribute value *v* in the concept hierarchy is calculated using the equation

$$IN(v,cht) = (k)d_{cht}(v) + (1 - k)wh_{cht}(v)$$

where *IN(v,cht)* is the interestingness of the attribute value *v* in the concept hierarchy tree *cht*, d_{cht} is the depth of *v* in the concept tree and wh_{cht} is the weighted height of *v* in the concept tree. The term *k* has a value ranging from 0 to 1 and is used to control the relative importance assigned to the depth and weighted height factors. Finally, the interestingness measure I_2 of the relation *R* is determined by the equation

$$I_2(R) = \sum_{(t,a,v) \in V_R} IN(v, tree(a))$$

where *tree(a)* yields the concept hierarchy for the attribute *a*.

These interestingness measures were used to compare the various selection strategies. The strategy that produced a final generalized relation with the largest interestingness measures is considered the most successful in discovering interesting information from the database.

3. ATTRIBUTE SELECTION STRATEGIES

3.1 Introduction

Currently, DB-Discover chooses the attribute in the prime relation that has the most distinct values for further generalization toward the final generalized relation. Strategies for selecting the

the attribute to generalize include selecting an attribute based on:

1. the largest reduction in the number of tuples [2,3,4].
2. the smallest reduction in the number of tuples [4].
3. the simplicity of the final generalized relation [2,3].
4. the maximum ratio of distinct attribute values to attribute threshold [5].

To illustrate the behavior of the strategies discussed in this section, a sample prime relation is shown in Table 1. The table contains attribute values for two attributes. Figure 1 depicts the upper parts of the concept hierarchies for these attributes. The concepts farthest from the roots of the trees represent the levels of generalization reached in the prime relation, with any less general nodes not shown. $T_a = 15$ for both attributes and $T_t = 10$ for the final generalized relation.

As required in a prime relation, the number of distinct attribute values for each attribute is less than or equal to its attribute threshold. However, the number of tuples in the relation exceeds the table threshold so further generalization is required. The further generalization of the prime relation towards the final generalized relation must be accomplished by generalizing one of the two attributes in the prime relation. Table 2 and Table 3 show tabular representations of the relations that would be obtained by generalizing attribute A and attribute B respectively.

3.2 Lookahead Strategies

In a *lookahead strategy*, each of the attributes is generalized in turn to create a new relation. Properties of these new generalized relations are used to chose the best attribute to use for the actual generalization. Only a one-step lookahead, where a decision is made after looking ahead one generalization, is considered here.

Attribute A	Attribute B
a15	b16
a16	b16
a17	b16
a18	b28
a19	b28
a20	b28
a23	b21
a24	b26
a27	b22
a28	b25
a31	b16
a31	b17
a31	b18

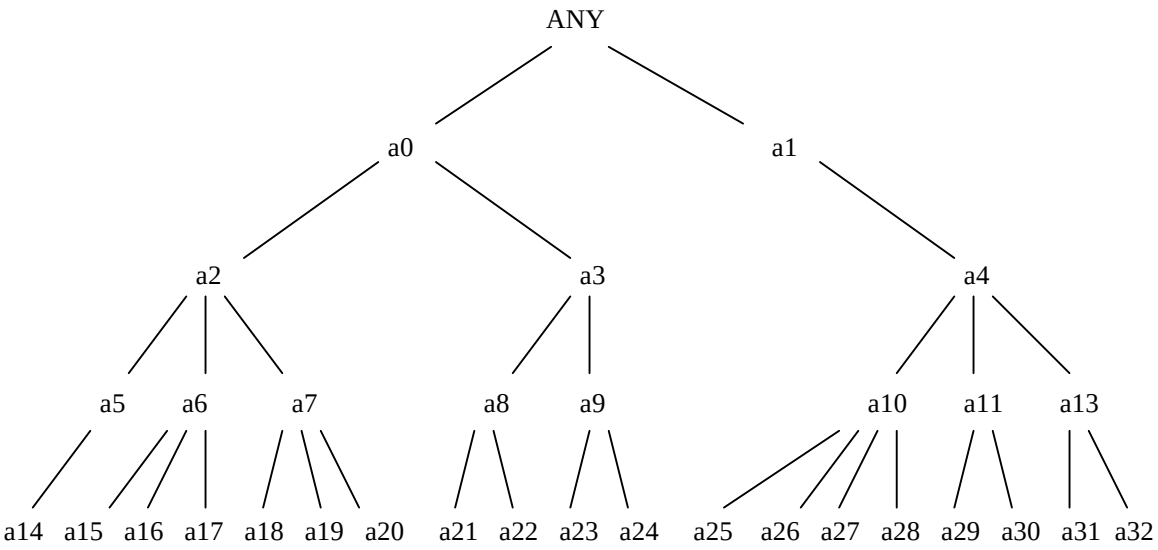
Table 1: Tabular Representation of the Sample Prime Relation

Attribute A	Attribute B
a6	b16
a7	b28
a9	b21
a9	b26
a10	b22
a10	b25
a13	b16
a13	b17
a13	b18

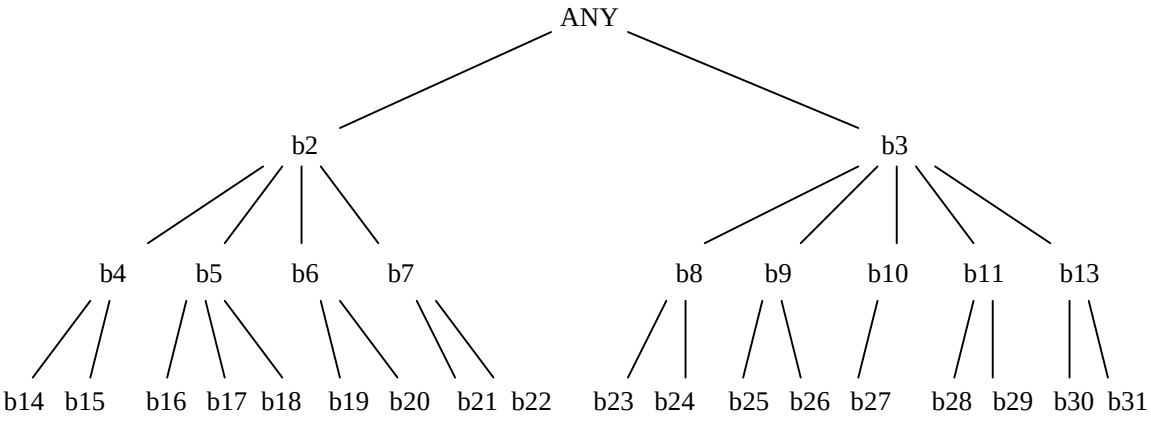
Table 2: Relation Obtained By Generalizing Attribute A

Attribute A	Attribute B
a15	b5
a16	b5
a17	b5
a18	b11
a19	b11
a20	b11
a23	b7
a24	b9
a27	b7
a28	b9
a31	b5

Table 3: Relation Obtained By Generalizing Attribute B



(a) Concept Hierarchy for Attribute A



(b) Concept Hierarchy for Attribute B

Figure 1: Concept Hierarchies for Sample Relation Attributes

3.2.1 Selection Based on the Number of Tuples After Generalization

Adopting the strategy of selecting an attribute whose generalization will result in the greatest reduction in tuples would likely result in a faster conversion of the prime relation to the final general relation. However, the final generalized relation is less likely to have properties close to the specified thresholds. For example, if the prime relation has close to the number of tuples specified by the table threshold, a large decrease in the number of tuples is not desirable. Not only could the number of tuples in the relation move farther from the table threshold, the number of distinct values for the attribute being generalized would move farther below the attribute threshold. The strategy of generalizing an attribute that yields the simplest resulting relation is likely to have similar properties, although a formal definition of the simplicity of a relation was not presented in [2][3].

Table 2 shows that a generalization of attribute A would reduce the number of tuples to 9. Table 3 shows that a generalization of attribute B would reduce the number of tuples to 11. Therefore, under this strategy, attribute A would be chosen.

A strategy based on the smallest decrease in the number of tuples would be more likely to result in a final generalized relation that has a number of tuples close to that specified for the table threshold and a number of distinct attribute values close to the specified attribute thresholds. However, the effort to attain that level of generalization will be higher.

In the sample prime relation, generalization of attribute B would produce a smaller reduction in the number of tuples than generalization of attribute A. Therefore, under this strategy, attribute B would be chosen.

Strategies involving the selection of the attribute that yields the greatest or smallest reduction in the number of tuples would require generalizing all attributes in the prime relation to find the attribute that meets the specified criteria. Most of the work involved in the generalization of the attributes does not directly contribute to the final generalized relation.

3.2.2 Selection Based on Interestingness Measures

Since one measure of the effectiveness of an attribute selection strategy is the interestingness measures discussed in the previous section, a lookahead strategy that produces the generalization to the most interesting relation should be effective. As with all lookahead methods discussed here, some of the work done in generalizing the relation using various attributes does not directly contribute to the final generalized relation.

To illustrate the behavior of the attribute selection strategy based on interestingness measures, it will be assumed that the concepts that are farthest from the root of the partial concept trees shown in Figure 1, are in the set V_R with $w(t) = 3$ and $n(t) = 3$. Also, it is assumed that $k = 0.5$ for the calculation of the interestingness of a node in a concept tree.

As an example of how the interestingness of each concept in a relation is determined, concept a6 in Table 2 will be examined. The concept is not a leaf concept or the most general concept ANY, so it is considered in the determination of the interestingness measures. There is a single instance of the concept in the relation represented in Table 2, so it adds 1 to the interestingness measure I_1. An examination of the concept hierarchy for attribute A shows that $d_{cht}(a6) = 3$ and that the concept a6 has children a15, a16 and a17. It has been assumed that each of these child nodes has $w(t) = 3$ and $n(t) = 3$. Based on this assumption, $n(a6) = 3 + 3 + 3 = 9$, $w(a6) = 9 + (3 + 3 + 3) = 18$ and $wht(a6) = 18/9 = 2$. Using k = 0.5, $IN(a6,cht) = 0.5(3) + 0.5(2) = 2.5$. Since there is a single instance of a6 in the relation represented in Table 2, it contributes 2.5 to the interestingness measure I_2.

For the generalized relation represented in Table 2, values used for the determination of interestingness measures I_1 and I_2 are summarized in Table 4. For the generalized relation represented in Table 3, values used for the determination of interestingness measures I_1 and I_2 are summarized in Table 5.

For the relation represented in Table 2, the interestingness measure $I_1 = 18$, the sum of the values in the instances column in Table 4. The interestingness measure $I_2 = 26.5$, the sum of the values obtained by multiplying the *IN* value of each concept by the number of instances of that concept. For the relation represented by Table 3, the interestingness measure $I_1 = 21$, the sum of the values in the instances column

Concept	Instances	d(t)	n(t)	w(t)	wh(t)	IN(t,cht)
a6	1	3	9	18	2	2.5
a7	1	3	9	28	2	2.5
a9	2	3	6	12	2	2.5
a10	2	3	12	24	2	2.5
a13	3	3	6	12	2	2.5
b16	2	3	3	3	1	2.0
b17	1	3	3	3	1	2.0
b18	1	3	3	3	1	2.0
b21	1	3	3	3	1	2.0
b22	1	3	3	3	1	2.0
b25	1	3	3	3	1	2.0
b26	1	3	3	3	1	2.0
b28	1	3	3	3	1	2.0

Table 4: Values For the Determination of Interestingness Measures for Table 2

Concept	Instances	d(t)	n(t)	w(t)	wh(t)	IN(t,cht)
a15	1	4	3	3	1	2.5
a16	1	4	3	3	1	2.5
a17	1	4	3	3	1	2.5
a18	1	4	3	3	1	2.5
a19	1	4	3	3	1	2.5
a20	1	4	3	3	1	2.5
a23	1	4	3	3	1	2.5
a24	1	4	3	3	1	2.5
a27	1	4	3	3	1	2.5
a28	1	4	3	3	1	2.5
b5	4	2	9	18	2	2.0
b7	2	2	6	12	2	2.0
b9	2	2	6	12	2	2.0
b11	3	2	6	12	2	2.0

Table 5: Values For the Determination of Interestingness Measures for Table 3

in Table 5. The interestingness measure I_2 = 47.0, the sum of the values obtained by multiplying the *IN* value of each concept by the number of instances of that concept.

I_1 for the relation obtained by generalizing attribute B is greater than I_1 for the relation obtained by generalizing attribute A. Similarly, I_2 for the relation obtained by generalizing attribute B is greater than I_2 for the relation obtained by generalizing attribute A. Therefore, under this strategy, attribute B would be chosen regardless of which interestingness measure was used to make the selection.

3.3 Predictive Strategies

In a *predictive strategy*, values that exist prior to the generalization are used to predict the best attribute to generalize.

3.3.1 Selection Based on the Number of Distinct Attribute Values

Methods that select the attribute in a relation that has the largest or smallest number of distinct attribute values (or ratio of the number of distinct attribute values to the attribute threshold), have the advantage of simplicity. The calculations are simple and no lookahead is involved. However, they do not consider the properties of the concept hierarchies or the actual generalization behavior. The use of the smallest number of distinct attribute values (or the smallest ratio of the number of distinct attribute values to the attribute threshold), essentially chooses an attribute for repeated generalization until the table threshold is met or until it has been maximally generalized. If the attribute is maximally generalized before the table threshold is reached, another attribute is chosen and the process repeated.

If the number of distinct values of an attribute *i* is represented by N_i and the ratio of distinct attribute values to attribute threshold is represented by $R_{N/Ta}$, then in the sample prime relation, N_A = 11 and N_B = 8. Since T_a = 15 for both attributes, $R_{N/Ta}$ = 0.733 for attribute A and $R_{N/Ta}$ = 0.533 for attribute B. If the largest *N* or $R_{N/Ta}$ were used as the selection criteria, then attribute A would be chosen for generalization. If the smallest *N* or $R_{N/Ta}$ were used as the selection criteria, then attribute B would be chosen for generalization.

3.3.2 Selection Based on the Complexity of the Remaining Concept Hierarchies

The interestingness based lookahead strategy, discussed in Section 3.2.2, converts the prime relation to a final generalized relation that contains as much detail as possible within the constraints specified by the user. However, it is desirable to eliminate the need to lookahead, since it involves a separate generalization of the relation being considered using each of the attributes in the relation. This can be done using a predictive technique that estimates the potential for interestingness of those concepts in the concept hierarchies that can be reached by ascending the concept hierarchy trees from the concepts in the relation being considered. The attribute whose remaining concept hierarchy has the greatest potential for interestingness is chosen for generalization.

Hamilton and Fudger [6] developed heuristic measures to estimate the potential for knowledge discovery from a database that are closely related to the interestingness measures I_1 and I_2. The measures are based on the complexity of the *concept forest*, a group of concept trees for the attributes in a database.

The first measure of the forest complexity, referred to as M_1, is simply the sum of the interior nodes in all of the concept trees in the forest. This is indicated by the equation

$$M_1(f) = \sum_{t \in f} nin_t$$

where t is a concept tree in the concept forest f and nin_t is the number of interior nodes in the tree.

The second measure of the complexity, referred to as M_2, is based on the number and interestingness of the interior nodes of the trees in the concept forest. This measure is determined by the equation

$$M_2 = \sum_{t \in f} \sum_{j \in Vt} IN(value(j,t),t)$$

where t is a concept tree in the concept forest f, j is a node in the concept tree, $value(j,t)$ is the concept value of the node and $IN(value(j,t),t)$ is the interestingness of the node.

Hamilton and Fudger [6] conducted experiments that indicated that there is good correlation between the complexity of the concept forest determined using the complexity measures M_1 and M_2 and the interestingness of the final general relation determined by the interestingness measures I_1 and I_2. The correlation was improved if only the concept trees relevant to the learning tasks were used.

Since the complexity measures M_1 and M_2 are closely related to the interestingness measures I_1 and I_2, it is logical to extend the use of the complexity measures to provide a guide for selecting an attribute in the prime relation for generalization toward the final general relation. However, the properties of the actual attribute values in the prime relation should be considered. An attribute in the prime relation will likely be at a level of generalization somewhere between the leaves and the root of the concept hierarchy. Only nodes in the concept tree with a depth less than that of the attributes in the current relation and greater than the depth of the root node should be considered, Additionally, branches for which there are no attributes values in the prime relation should be removed.

For attribute A, the concepts a0, a1, a2, a3, a4, a6, a7, a9, a10 and a13 would be considered. These are the concepts that are not the concept ANY or that have more specific concept values in the prime relation that lead to them as the tree is ascended. The concepts in the hierarchy tree for attribute B that meet these criteria are b2, b3, b5, b7, b9 and b11. The concepts in the relation being considered are treated as leaf nodes, with $w(t) = 0$ and $n(t) = 1$. Values for the interestingness of a concept *IN* are determined using $k = 0.5$. Values used for the determination of complexity measures M_1 and M_2 for attribute A are summarized in Table 6. Values used for the determination of complexity measures M_1 and M_2 for attribute B are summarized in Table 7.

For the remaining hierarchy tree of attribute A, the complexity measure $M_1 = 10$, the number of concepts reachable from the current concept values, excluding the most general concept ANY. The complexity measure $M_2 = 20$, the sum of the *IN* column in Table 6. For the remaining hierarchy tree of attribute B, the complexity measure $M_1 = 6$, the number of concepts reachable from the current concept values, excluding the most general concept ANY. The complexity measure $M_2 = 9$, the sum of the *IN* column in Table 7.

Concept	d(t)	n(t)	w(t)	wh(t)	IN(t,cht)
a0	1	8	24	3	2.0
a1	1	3	9	3	2.0
a2	2	6	12	2	2.0
a3	2	2	4	2	2.0
a4	2	3	6	2	2.0
a6	3	3	3	1	2.0
a7	3	3	3	1	2.0
a9	3	2	2	1	2.0
a10	3	2	2	1	2.0
a13	3	1	1	2	2.0

Table 6: Values Used For the Determination of Complexity Measures for Attribute A

Concept	d(t)	n(t)	w(t)	wh(t)	IN(t,cht)
b2	1	5	10	2	1.5
b3	1	3	6	2	1.5
b5	2	3	3	1	1.5
b7	2	2	2	1	1.5
b9	2	2	2	1	1.5
b11	2	1	1	1	1.5

Table 7: Values Used For the Determination of Complexity Measures for Attribute B

M_1 for the remaining concept hierarchy of attribute A is greater than M_1 for the remaining concept hierarchy of attribute B. Similarly, M_2 for the remaining concept hierarchy of attribute A is greater than M_2 for the remaining concept hierarchy of attribute B. Therefore, under this strategy, attribute A would be chosen regardless of which complexity measure was used to make the selection.

This is opposite to the results that would be obtained using the lookahead strategy based of the interestingness of the generalized relation, where attribute B was chosen for generalization. Although this may seem odd, it simply means that for the next level of generalization, a generalization of attribute B gives the most interesting relation. The choice of attribute A is based upon the potential for interestingness of the remaining portions of the hierarchy trees.

3.4 Time Complexity Analysis of a Single Generalization Step

3.4.1 Lookahead Strategies

In the lookahead strategies, the input relation must be generalized using each of its n attributes. Each of the m tuples in the input relation must be examined, the appropriate concept generalized one level to create a new tuple and the new tuple inserted into the new relation, if it not already there. The cost to carry out the actual generalization by tree ascension is relatively small for large input relations compared to the cost to create the new relation [8]. A new relation is constructed by inserting any unique new tuples into the relation. To determine if a tuple is unique, it is compared to the tuples previously inserted in the relation. If the new relation has p tuples, the comparison and insertion of each tuple would take at most np steps. The creation of the new relation would take at most npm steps. Therefore, the generalization of a single attribute to create a new relation is O(npm). The time complexity to carry out this generalization and relation creation for n attributes is O(n^2pm) since only one attribute is generalized to create each new relation.

In the lookahead strategies based on the number of tuples in the generalized relation, no further work is required except the comparison of the number of tuples of the new relations. This is done as the new relations are created so the time complexity of these strategies is O(n^2mp).

For the lookahead strategy based on the interestingness measures I_1 and I_2, the interestingness of the n generalized relations must be determined. This involves examining each concept in a relation and calculating the interestingness of any concept that is not the most general concept ANY or a leaf concept. The process of determining which concepts contribute to the interestingness of the relation can be done using existing information associated with the concepts in the relation, so np steps are required for each attribute or n^2p steps for the n new generalized relations.

For the interestingness measure I_1, a count of the concepts that contribute to the relation interestingness is sufficient to make a selection, so the complexity of this strategy is O($n^2mp + n^2p$) or O(n^2pm). For the strategy based on interestingness measure 1_2, some additional work is required before the interestingness of the concepts can be calculated. Each concept in the concept hierarchies associated with the attributes in the input relation must be visited prior to the attribute selection to determine the

values needed for the calculation of I_2. However, if this tree traversal was done only once prior to starting the further generalization of the prime relation, it can be removed from consideration of the time complexity of a single generalization. The time complexity of this strategy also becomes O(n^2pm).

3.4.2 Predictive Strategies

In the predictive strategies, some property of the input relation is used to determine what attribute to generalize. For the strategies based on the number of distinct attribute values or $R_{N/Ta}$ ratios, the information needed to select an attribute has already been determined and is associated with the input relation. The time complexity to determine the maximum or minimum value of the property being considered is O(n).

For the strategies based on the complexity measures of the remaining hierarchy trees, more work is required. For each of n attributes in the input relation, a list of the distinct concept values in the relation is created. This involves examining each of the m concepts in the input relation for an attribute, checking if the concept is already in the list of distinct concepts and if not, inserting the concept into the list. The linear search of the list would examine at most m list entries. The insertion of a concept into the list is O(1), since it is always inserted at the end of the list. Therefore, the search and insertion of a distinct concept is O($m + 1$) or O(m). The consideration of a non-unique concept is also O(m). To create the list for all attributes is O(nm). Then for each concept in a list, the path to the most general concept ANY is traveled, inserting any new distinct interior concepts encountered into a list of more general concepts remaining in the hierarchy tree for that attribute. If there are m_d concepts in the initial list and h concepts in the concept hierarchy tree, then at most (h - m_d) concepts may need to be examined and inserted in a list. The time complexity of this operation is O((h - m_d) + 1) or O(h - m_d). The time complexity to creating both lists for an attribute is O(m + (h - m_d)) and for all attributes, O(nm +n(h-m_d)) or O(nm + nh).

For complexity measure M_1, a count of the interior concepts in the remaining concept hierarchies is sufficient to make a selection, so the time complexity of this strategy is O(nm + nh). For the strategy based on interestingness measure M_2, some additional work is required. Each of the involved

concepts trees must be traversed to determine the values needed to calculate the complexity measure M_2. If a concept hierarchy contains h concepts, than the concept tree traversal is O(h) and for all attributes in the input relation O(nh). This must be done for each generalization step since only the remaining concepts contribute to the complexity measure. The time complexity of this strategy becomes O($nm + nh + nh$) or O($nm + nh$).

4. STRUCTURE OF THE DB-DISCOVER PROGRAM.

The DB-Discover program is functionally divided into 5 main modules [4]:

1. User Interface Module
2. Command Module
3. Database Access Module
4. Concept Hierarchy Module
5. Learning Module

The primary module involved in the generalization activities is the learning module. It receives as input the relation to be generalized, a set of concept hierarchy trees used to guide the generalization and a structure containing parameters that affect the generalization procedures. The learning process consists of the initial generalization of the attributes, removal of duplicate tuples and the final generalization. Although the module previously used a single strategy for the generalization from the prime relation to the final general relation, it was designed to allow the use of different attribute selection strategies. This includes routines to set the strategy to be used, determine what strategy is being used and to call the appropriate routine to carry out the generalization. Routines to implement the various attribute selection strategies were placed in a submodule and linked to the learning module.

5.0 EXPERIMENTAL PROGRAM

The experimental data was obtained using a remote version of the DB-Discovery program. The client machine was a Silicon Graphics workstation, running an IRIX Release 5.3 operating system, accessed through an IBM compatible 486 personal computer with an OS/2 version 3.0 operating system. An X-Windows interface to the DB-Discover program was employed. The server portion of the remote version of DB-Discover was installed on a Sun IPX workstation, running a Sun Release 4.1.3 operating system. DB-Discover accessed the database through an Oracle Version 7.0.1.16 DBMS.

5.1 Implemented Strategies

A number of selection strategies were implemented, including both backtracking and predictive strategies. The implemented predictive strategies include selection based on the:

1. least number of distinct attribute values.
2. largest ratio of the number of distinct attribute values to attribute threshold.
3. smallest ratio of the number of distinct attribute values to attribute threshold.
4. largest complexity measure M_1.
5. largest complexity measure M_2

Test results were also obtained for the previously implemented strategy based on the greatest number of distinct attribute values. The implemented lookahead strategies include selection based on the:

1. greatest reduction in the number of tuples.
2. least reduction in the number of tuples.
3. largest interestingness measure I_1.
4. largest interestingness measure I_2.

5.2 Description of Experiments

At the time of this paper, only a single database could be accessed through the remote version of DB-Discover. The database consists of National Engineering and Science Research Council of Canada (NSERC) grant and award information. It is a relatively small 10,000 tuple database with only three

concept hierarchies suitable for adequately examining attribute selection strategies. Most of the concept hierarchies are wide and shallow. Additionally, the hierarchies have numerous one-to-one generalizations, where a single concept value is generalized to another single concept value. Such hierarchy trees have a limited potential for interesting discoveries [6]. However, by using learning tasks that included the attributes that do have relatively complex hierarchy trees, expanding other hierarchies and setting the attribute thresholds to relatively high values, sufficient information was obtained to compare the performance of the implemented strategies.

The experimental program consisted of the step-by-step generalization of four prime relations to final generalized relations using each of ten implemented attribute selection strategies. The four prime relations were obtained from the database using three different learning task specifications. Test Series 1 involved four attributes AMOUNT, AREA_CODE, DISC_CODE and PROVINCE. Test Series 2 involved the same four attributes, but a more complex concept hierarchy was used for the attribute DISC_CODE. Test Series 3 involved only the attributes AMOUNT, DISC_CODE and PROVINCE. Test Series 4 again involved all four attributes, but a more complex concept hierarchy was used for the attribute AREA_CODE. Information about the learning tasks that were used to obtain the prime relations and the concept hierarchies that were used to reduce the relations are contained in Appendix A.

Once the prime relation was obtained, it was repeatedly generalized using each of the implemented attribute selection strategies. After each generalization step, the interestingness measures I_1 and I_2 were computed for the new generalized relation. The generalization was continued until the number of tuples in the generalized relation was less than a table threshold of 10. In all, information was obtained for 428 relations.

6. EXPERIMENTAL RESULTS

For compactness, the strategies will be referenced in all tables and charts according to the reference codes in Table 8.

Attribute Selection Strategy	Strategy Code
Most Distinct Attribute Values	s1
Greatest Reduction in Tuples	s3
Least Reduction in Tuples	s4
Largest Complexity Measure M_1	s5
Largest Complexity Measure M_2	s6
Greatest $R_{n/Ta}$	s7
Least $R_{n/Ta}$	s8
Least Distinct Attribute Values	s9
Largest Interestingness Measure I_1	s10
Largest Interestingness Measure I_2	s11

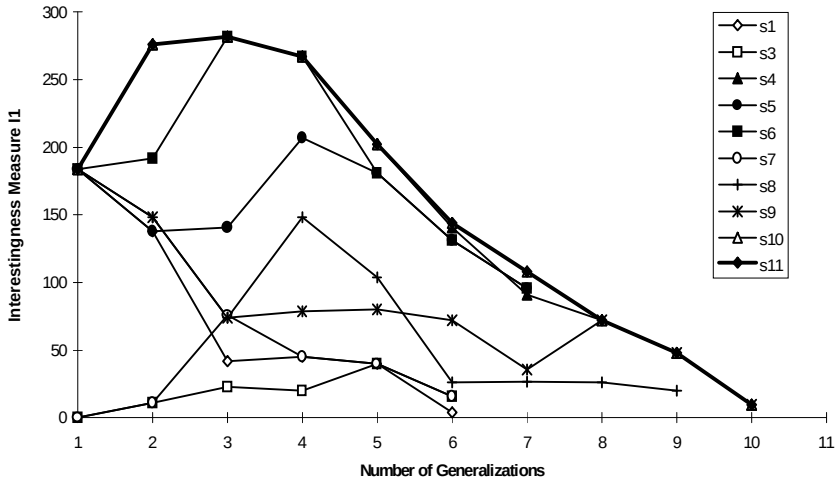
Table 8: Attribute Selection Strategy Reference Codes

The interestingness measures calculated for each of the relations obtained during the step-by-step generalization of the prime relation towards the final generalized relation are presented graphically in Figure 2 for Test Series 1, Figure 3 for Test Series 2, Figure 4 for Test Series 3 and Figure 5 for Test Series 4. Each figures consists of two charts, one for Interestingness Measure I_1 and one for interestingness measure I_2. The measures are plotted against the number of generalizations from the prime relation toward the final generalized relation. The same information is displayed in tabular form in Appendix B.

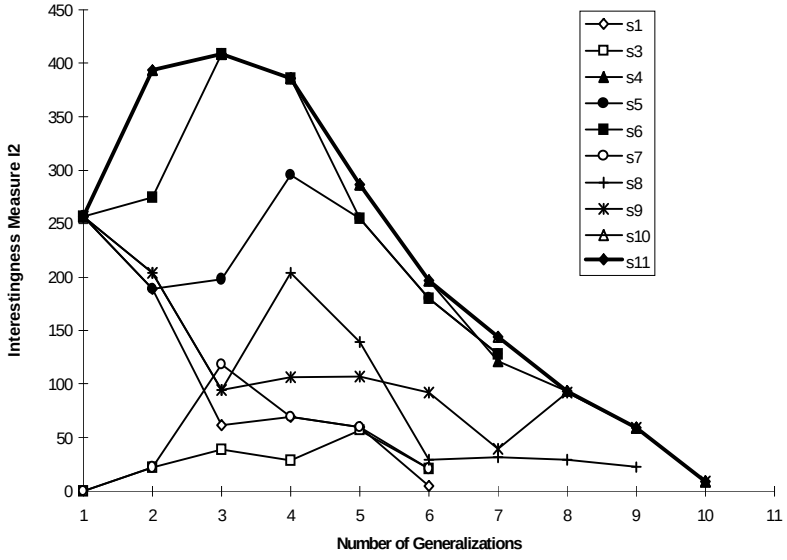
6.1 Discussion of Results

In each test series, the magnitudes and behavior of the calculated interestingness measures suggests that the strategies can be loosely clustered into two groups. For the purposes of discussion these will be referred to as Group 1 and Group 2. *Group 1* consists of the lookahead strategies based on the interestingness measures I_1 and I_2, the lookahead strategy based on the least reduction in the number of tuples and the predictive strategies based on the complexity measures M_1 and M_2. *Group 2* consists of the lookahead strategy based on the greatest reduction in the number of tuples, the predictive strategies based on the number of distinct attribute values and the predictive strategies that are based on the $R_{N/Ta}$ ratios.

The lookahead strategies based on interestingness measures almost always produced the relations with highest interestingness at a given number of generalization steps. However, a few

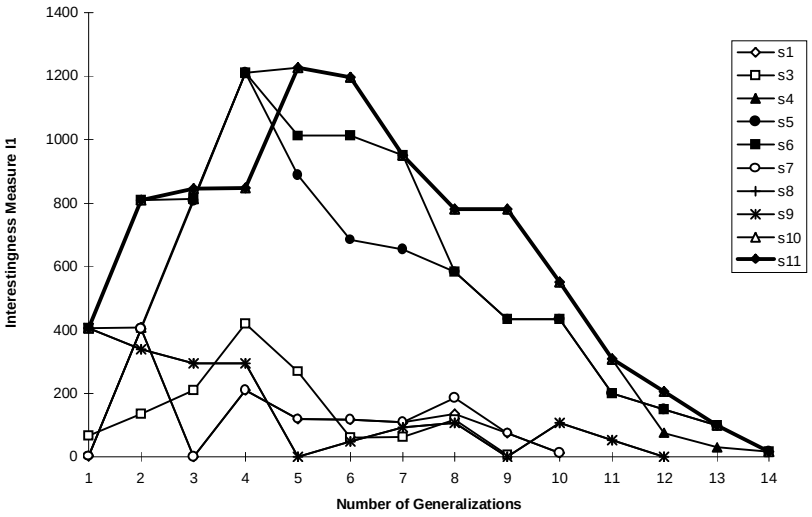


a) Interestingness Measure I_1

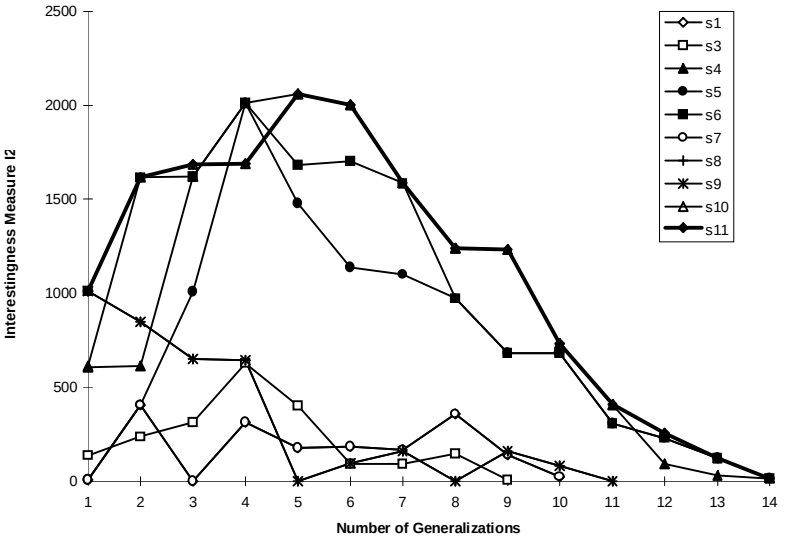


b) Interestingness Measure I_2

Figure 2: Interestingness Measures Versus Number of Generalizations for Test Series 1

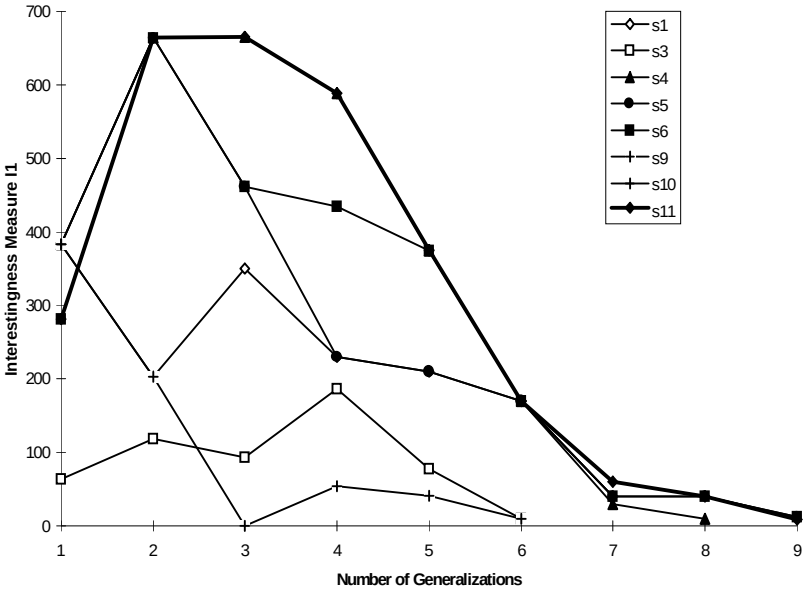


a) Interestingness Measure I_1

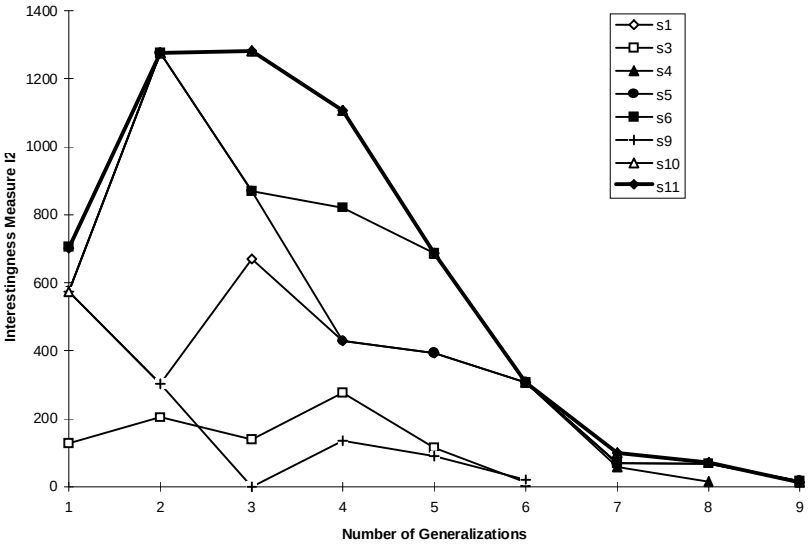


b) Interestingness Measure I_2

Figure 3: Interestingness Measures Versus Number of Generalizations for Test Series 2

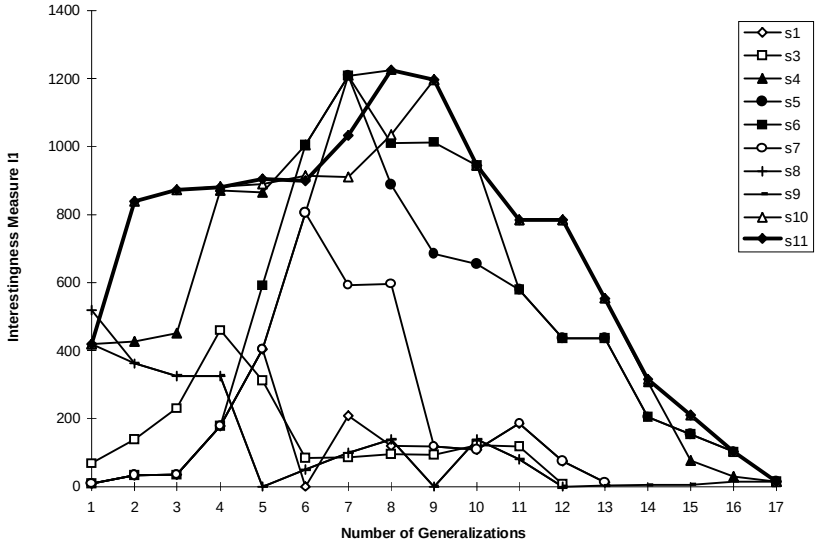


Interestingness Measure I_1

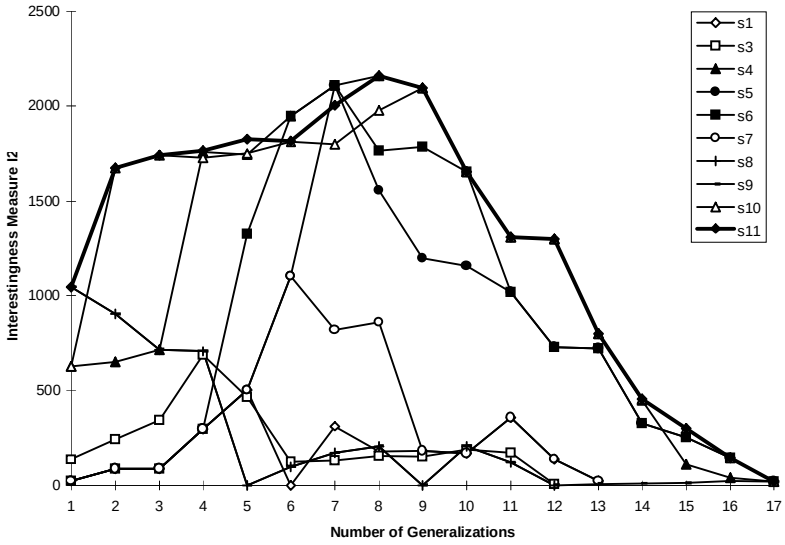


b) Interestingness Measure I_2

Figure 4: Interestingness Measures Versus Number of Generalizations for Test Series 3



a) Interestingness Measure I_1



b) Interestingness Measure I_2

Figure 5: Interestingness Measures Versus Number of Generalizations for Test Series 4

exceptions to this were noted. This is illustrated in Figure 3 at generalization step 4 and Figure 5 at generalization steps 6 and 7. Here, the other strategies belonging to Group 1 produced more interesting relations. An examination of the relations prior to this generalization step shows that this occurred, at least in part, due to the way DB-Discover carries out the generalization of an attribute in a relation that has instances of concepts from different depths in the concept tree. Each generalization step is carried out only for the attribute concepts at the greatest depth in the concept tree. If an attribute in a relation has a concept hierarchy with a small number of concepts at a lower level and there are one or more instances of these concepts in the relation, the generalization of that attribute would cause only a small increase in the interestingness of the relation. Such an attribute was chosen for generalization at an earlier generalization step by the other strategies in Group 1. In the generalizations that produced the anomalies in interestingness, that same attribute was chosen by both interestingness based lookahead strategies. Since this attribute had already been generalized by the other strategies in the group, they selected a different attribute that gave a much higher increase in the interestingness. In spite of these anomalies, the interestingness measures produced using the lookahead strategies based on interestingness measures would appear to provide some indication of an upper limit of the interestingness measures that may result during the further generalization of a particular prime relation.

To provide an approximate ranking of the strategies in terms of their ability to produce interesting results, the interestingness measures produced by strategy s11 was compared to the interestingness measures produced by the other strategies. The comparison was done by first calculating the average percent difference in the interestingness measures for a given test series, then averaging these averages over the four test series. The results are shown in Table 9 on the following page.

The interestingness measures of the relations produced by the Group 1 strategies are consistently higher than the interestingness measures of the relations produced by the Group 2 strategies. For the most part, this result was expected. Group 1 includes the lookahead strategies based on the interestingness measures, which consider the actual interestingness produced by a generalization and the predictive strategies based on the complexity measures, which consider the potential for interestingness that could result from a generalization. None of the strategies in Group 2 explicitly consider these measures at all.

		Average % Difference from Interestingness Measure I_2 for s11								
Strategy Code	Group	I_1, Test Series 1	I_2, Test Series 1	I_1, Test Series 2	I_2, Test Series 2	I_1, Test Series 3	I_2, Test Series 3	I_1, Test Series 4	I_2, Test Series 4	
s11	1	-	-	-	-	-	-	-	-	
s10	1	0	0	0	-3	-2	-2	-2	-4	
s4	1	0	-2	-10	-16	-17	-17	-12	-17	
s6	1	-10	-9	-11	-9	-6	-6	-33	-31	
s5	1	-20	-23	-26	-29	-17	-17	-40	-41	
s8	2	-33	-57	-77	-76	-	-	-76	-77	
s9	2	-30	-39	-77	-76	-77	-77	-77	-76	
s1	2	-66	-66	-86	-88	-39	-39	-88	-89	
s7	2	-87	-86	-86	-88	-	-	-75	-80	
s3	2	-92	-91	-83	-86	-85	-84	-83	-86	

Table 9: Ranking of the Effectiveness of a Strategy in Producing Interesting Results

One unexpected result was the effectiveness in producing interesting results of the strategy based on choosing the attribute that will produce the least reduction of tuples. Each instance of a concept in a relation that corresponds to an interior concept in the concept hierarchy tree contributes to the total interestingness of that relation. A relation with more tuples is likely to have more concepts that contribute to that total. This strategy appears to be almost as effective as those that directly consider the interestingness or complexity measures.

Although the Group 1 strategies produce more interesting relations, they require more generalization steps to reach the final generalized relation. As a rough estimate, the Group 1 strategies require approximately 30% more generalization steps than the Group 2 processes. Since the Group 2 strategies have been shown to be generally more efficient than those in Group 1 for each generalization step, there may be a significant trade-off between the interestingness of the generalization results and the efficiency of the strategies in producing those results. The exception to this in Group 2 is the strategy for selecting the attribute that results in the largest tuple reduction, which is neither efficient nor effective.

The interestingness measures of the Group 1 strategies increase initially. After the interestingness reaches a maximum value, it decreases with increased generalization toward the final generalized relation. The initial increase can be attributed to the relatively high attribute thresholds that were specified, resulting in prime relations consisting primarily of leaf concepts, and the gradual generalization processes of the Group 1 strategies. As the initial generalization steps occur, leaf concepts are gradually replaced by more interesting interior concepts. The interestingness increases because the gain in interestingness due to the generalization of leaf concepts is greater than the loss of interestingness due to redundant tuple elimination and generalization of concepts to the most general concept ANY. Eventually, most of the leaf concepts are generalized and the loss in interestingness exceeds the gain in interestingness, resulting in a net interestingness decrease. The interestingness of the Group 2 strategies generally display a much smaller initial increase in interestingness or no increase at all. It appears that the faster or more coarse generalization strategies dampens or eliminates this affect.

For both groups, the variation in the interestingness between members of the group is greatest during the initial generalization steps. The variation decreases as the generalization process proceeds

toward the final generalized relation. This may be the result of the characteristics of the different concept hierarchies or of the attribute selection strategies themselves. One example that can be attributed to the characteristics of the strategies, is the convergence of the interestingness produced by the lookahead interestingness based strategies and the interestingness produced by the predictive complexity based strategies. The interestingness based strategies consider the interestingness of each instance of an interior concept in the generalized relation. The complexity based strategies consider the potential for interestingness of the single instances of the interior concepts in the remaining hierarchy tree. As the generalization level increases, the number of instances of particular concepts will decrease and there will be fewer concepts in the remaining hierarchy tree. Therefore, there is a better chance that the concepts considered by both methods would be the same.

The use of the lookahead strategy based on interestingness measure I_1 is generally as effective as the strategy based on interestingness measure I_2 in producing interesting results. For the most part, each strategy chooses the same attribute for further generalization. This is also indicated in the values of the interestingness measures calculated for the generalized relations. The graphical representation of the results in Figures 2, 3, 4 and 5 show that the measures differ in magnitude but not in behavior. It is possible that any additional effort required to calculate I_2 may not be warranted in this application. This does not appear to be true for the complexity based strategies. The strategy based on the complexity measure M_2 was approximately twice as effective than the strategy based on complexity measure M_1.

7.0 CONCLUSIONS AND RECOMMENDATIONS

One step in the process of the attribute-oriented generalization performed by DB-Discover is the generalization that must be done if the number of tuples in the prime relation exceeds the specified table threshold. Currently, DB-Discover chooses the attribute with the most distinct values for further generalization. This paper describes the implementation and comparison of a number of different attribute selection strategies suggested in the literature [2], [3], [4], [5] and additional strategies based on the interestingness and complexity measures developed by Hamilton and Fudger [6] . The criteria for

evaluating the strategies was the ability of a strategy to produce interesting results and the efficiency of the strategy in producing those results.

The most effective attribute selection strategies for producing interesting results were the lookahead strategies based on selecting the attribute that produced the highest interestingness measures I_1 and I_2. They consistently produced the most interesting relations. They are effective because they directly consider the interestingness measures used to evaluate this aspect of the attribute selection strategy performance. The strategy based on I_1 was found to be as effective as the strategy based on I_2, indicating that any additional effort needed to compute I_2 is not warranted in this application.

The lookahead strategy based on selecting the attribute that produced the least reduction in the number of tuples in the generalized relation proved to be almost as effective in producing interesting results as the interestingness based strategies. This can be attributed to the minimized loss in interestingness due to redundant tuple elimination in the generalization process.

The strategies based on selecting the attribute whose concept hierarchy had the most potential for interestingness, as measured by the complexity measures M_1 and M_2, were less effective in producing interesting results for the first few generalization steps from the prime relation. However, as the generalization process proceeded, the interestingness of the relations produced by these strategies converged with those of the lookahead strategies previously described. These strategies are effective because they consider the potential for interestingness in the remaining portions of the involved concept hierarchies. The strategy based on the complexity measure M_2 was found to be approximately twice as effective in producing interesting results than the strategy based on the complexity measure M_1. Since both have the same time complexity, the use of M_2 in the selection process is preferable.

The least effective strategies in producing interesting results were found to be the predictive strategies that selected the attribute with the most or least number of distinct values, the predictive strategies that selected the attribute with the largest and smallest ratio $R_{N/Ta}$ and the lookahead strategy that selected the attribute that resulted in the largest reduction in the number of tuples in the generalized relation. These strategies do not explicitly consider the interestingness of the results.

An analysis of the time complexity of the different attribute selection strategies shows that in general, the strategies that produce the most interesting results are the least efficient in doing so. Not only are they less efficient in each generalization step, they were found to require approximately 30% more steps to reach a relation that satisfied the table threshold requirement.

The results of these experiments show that in selecting the best strategy for the further generalization from the prime relation to the final generalized relation, there is a trade-off between the interestingness of the relation that is produced and the efficiency with which it will be produced. However, since the generalization will usually be done on an already generalized prime relation and the smaller further generalized relations leading to the final generalized relation, the efficiency of a selection strategy may be somewhat less important than the effectiveness of the strategy in producing interesting results. For this reason, the lookahead strategy based on the interestingness measure I_1, the lookahead strategy based on the least tuple reduction or the predictive strategy based on the complexity measure M_2 would seem to be the most suitable.

7.0 REFERENCES

[1] W. J. Frawlry, G. Piatetsky-Shapiro and C. J. Matheus, "Knowledge Discovery in Databases:An Overview," in: G. Piatetsky-Shapiro and W. J. Frawly, Eds., *Knowledge Discovery in Databases*, AAAI/MIT Press, Menlo Park, CA, 1991, Pages 1-27.

[2] Y. Cai, N. Cercone and J. Han, "Attribute-oriented Induction in Relational Databases," in: G. Piatetsky-Shapiro and W. J. Frawly, Eds., *Knowledge Discovery in Databases*, AAAI/MIT Press, Menlo Park, CA, 1991, 213-228.

[3] J. Han, Y. Cai and N. Cercone, "Knowledge Discovery in Databases: An Attribute-Oriented Approach," *Proceedings of the 18th VLDB Conference*, Vancouver, British Columbia, 1992, 547-559.

[4] C. L. Carter, H. J. Hamilton and N. Cercone, "*The Software Architecture of DBLEARN*, Technical Report CS-94-04," University of Regina, 1994.

[5] N. Shan, H. J. Hamilton and N. Cercone, "GRG: Knowledge Discovery Using Information Generalization, Information Reduction and Rule Generation," *7th IEEE International Conference on Tools with Artificial Intelligence*, Washington, D.C., November, 1995, Accepted.

[6] H. J. Hamilton and D. F. Fudger, "Estimating DBLEARN's Potential for Knowledge Discovery in Databases," *Computational Intelligence*, **11**(2), 1995.

[7] C. B. Rivera and C. L. Carter, "*A Tutorial Guide to DB-Discover, Version 2.0*, Technical Report CS-95-05," University of Regina, 1995., 280-296

[8] C. L. Carter, "Efficient Attribute-oriented Generalization for Knowledge Discovery From Large Databases", M.Sc. Thesis, 1994

APPENDIX A - Learning Tasks and Concept Hierarchies

A1. Learning Tasks

Three different learning tasks were used to obtain prime relations. Task 1 was used in Test Series 1. Task 2 was used in Test Series 2 and Test Series 4. Task 3 was used in Test Series 3.

A1.1 Task 1

Task Type:	Characteristic
Hierarchy File:	nserc.chf
Selected Relations:	AWARD, ORGANIZATION
Join Tables:	AWARD A, ORGANIZATION B
Join Attributes:	Relation 1: ORG_CODE, Relation 2: ORG_CODE
Attributes Selected:	AMOUNT, AREA_CODE, DISC_CODE, PROVINCE
Attribute Qualifications:	DISC_CODE = "HARDWARE", DISC_CODE = "SOFTWARE"
Attribute Thresholds:	DEFAULT T_a = 10, AMOUNT T_a = 15, AREA_CODE T_a = DEFAULT, DISC_CODE T_a = DEFAULT, PROVINCE T_a = 12

A1.2 Task 2

Task Type:	Characteristic
Hierarchy File:	nsercmod1.chf for Test Series 2 nsercmod2.chf for Test Series 4
Selected Relations:	AWARD, ORGANIZATION
Join Tables:	AWARD A, ORGANIZATION B
Join Attributes:	Relation 1: ORG_CODE, Relation 2: ORG_CODE
Attributes Selected:	AMOUNT, AREA_CODE, DISC_CODE, PROVINCE
Attribute Qualifications:	DISC_CODE = "STRUCTURAL_ENGINEERING", DISC_CODE = "MECHANICAL ENGINEERING"
Attribute Thresholds:	DEFAULT T_a = 20, AMOUNT T_a = DEFAULT, AREA_CODE T_a = DEFAULT, DISC_CODE T_a = DEFAULT, PROVINCE T_a = DEFAULT

A1.3 Task 3

Task Type:	Characteristic
Hierarchy File:	nsercmod1.chf
Selected Relations:	AWARD, ORGANIZATION
Join Tables:	AWARD A, ORGANIZATION B
Join Attributes:	Relation 1: ORG_CODE, Relation 2: ORG_CODE
Attributes Selected:	AMOUNT, DISC_CODE, PROVINCE
Attribute Qualifications:	DISC_CODE = "PHYSICS AND CHEMISTRY"
Attribute Thresholds:	DEFAULT T_a = 20, AMOUNT T_a = DEFAULT, AREA_CODE T_a = DEFAULT, DISC_CODE T_a = DEFAULT, PROVINCE T_a = DEFAULT

A2 Concept Hierarchies

Three different concept forests were used to guide the generalizations of the attributes in the prime relation. The concept trees for the attributes AMOUNT and PROVINCE remained the same in all forests. The concept tree for the attribute DISC_CODE was modified for Test Series 2, Test Series 3 and Test Series 4. The Concept tree for the attribute AREA_CODE was modified for Test Series 4.

The concept trees for the relevant attributes are show below in the tabbed format used by DB-Discover. Each tab represents an increase in the depth of the concept by one.

A.2.1 Initial Concept Hierarchy

province
- Canada
  - Ontario
  - Quebec
  - Western
    - British Columbia
    - Prairies
      - Alberta
      - Saskatchewan
      - Manitoba
  - Atlantic
    - New Brunswick
    - Nova Scotia
    - Newfoundland
    - PEI
  - Other in Canada
- Outside Canada

area_code
- Agricultural
  - 100~200
- Computer_Science
  - Computer_Software
    - 860~870
  - Computer_Hardware
    - 870~880
  - Other_in_Computer
    - 850~860
- Energy
  - 200~300
- Environment
  - 300~400
- Earth
  - 400~500
- Human_Health
  - 500~550
- Education_Business
  - 550~600
- Construction_Tech
  - 600~700
- Industrial
  - 700~800
- Material
  - 800~850
  - 880~900
- Trans_Telecomm
  - 900~1000
- Space_Aeronomy
  - 1000~1100
- Northern_Develop
  - 1100~1200
- Knowledge
  - 1200~1210
- Other
  - 0~100
  - 1210~10000

- amount
  - 0-20Ks
    - 0-10Ks
      - 0~10000
    - 10Ks-15Ks
      - 10000~15000
    - 15Ks-20Ks
      - 15000~20000
  - 20Ks-40Ks
    - 20Ks-25Ks
      - 20000~25000
    - 25Ks-30Ks
      - 25000~30000
    - 30Ks-40Ks
      - 30000~40000
  - 40Ks-60Ks
    - 40Ks-50Ks
      - 40000~50000
    - 50Ks-60Ks
      - 50000~60000
  - 60Ks-
    - 60Ks-100Ks
      - 60000~100000
    - 100Ks-
      - 100000~100000000

- disc_code
  - Computer
    - HARDWARE
      - 23000~23500
    - SYS_ORGANIZATION
      - 23500~24000
    - SOFTWARE
      - 24000~24500
    - THEORY
      - 24500~25000
    - MATHEMATICS
      - 25000~25500
    - DATABASES
      - 25500~26000
    - AI
      - 26000~26500
    - COMPING_METHODS
      - 26500~27000
  - Other
    - 0~23000
    - 27000~70000

A.2.2 Modified Concept Tree for the DISC_CODE Attribute

disc_code
- Engineering_and_Computing_Science
  - Engineering
    - Structural_engineering
      - CONSTRUCTION_ENGINEERING_AND_MANAGEMENT
        - 00500~01000
      - STRUCTURAL_MATERIALS
        - 01000~01500
      - STRUCTURAL_ENGINEERING
        - 01500~02000
    - SURVEYING_ENGINEERING
      - 02000~02500
    - GEOTECHNICAL_ENGINEERING
      - 02500~03000
    - TRANSPORTATION_ENGINEERING_AND_PLANNING
      - 03000~03500
    - AEROSPACE_AND_AERONAUTICAL_ENGINEERING
      - 03500~04000
    - HYDRAULIC_ENGINEERING
      - 04000~04500
    - COASTAL_LAKES_AND_RIVER_ENGINEERING
      - 04500~05000
    - OFFSHORE_ENGINEERING
      - 05000~05500
    - Renewable_resources
      - AGRICULTURAL_ENGINEERING
        - 05500~06000
      - FOREST_ENGINEERING
        - 06000~06500
    - ENVIRONMENTAL_ENGINEERING
      - 06500~07000
    - Mechanical_engineering
      - FLUID_MECHANICS
        - 07000~07500
      - MECHANICS
        - 07500~08000
      - MECHANICAL_SYSTEMS_AND_INSTRUMENTATION
        - 08000~08500
    - DESIGN_AND_MANUFACTURING
      - 08500~09000
    - Thermal_engineering
      - ENGINEERING_THERMODYNAMICS
        - 09000~09500
      - HEAT_TRANSFER
        - 09500~10000
      - COMBUSTION
        - 10000~10500
    - PRODUCTION_AND_OPERATIONS_MANAGEMENT
      - 10500~11000
    - OPERATIONS_RESEARCH
      - 11000~11500
    - Non_renewable_resources
      - NON_RENEWABLE_RESOURCE_MANAGEMENT
        - 11500~12000
      - MINING_AND_MINERAL_PROCESSING
        - 12000~12500
    - Minerals_and_metallurgy
      - MATERIALS_PROCESSING_AND_METALLURGY
        - 12500~13000
      - MATERIALS_AND_METALLURGICAL_STRUCTURE_AND_PROPERTIES
        - 13000~13500
      - SPECIFIC_AND_METALLURGICAL_MATERIALS
        - 13500~14000
    - Chemical_processes
      - REACTION_FUNDAMENTALS_AND_REACTOR_DESIGN
        - 14000~14500
      - SEPARATION_PROCESSESS
        - 14500~15000

TRANSPORT_PROCESSES
15000~15500
OCCUPATIONAL_AND_ENVIRONMENTAL_SAFETY
15500~16000
ENERGY_CONVERSION_AND_UTILIZATION_PROCESSESS
16000~16500
NUCLEAR_ENGINEERING
16500~17000
PROCESS_OPTIMIZATION_AND_CONTROL
17000~17500
SPECIFIC_INDUSTRIAL_PROCESSES
17500~18000
PHOTON_DEVICES
18000~18500
ELECTROMAGNETICS
18500~19000
COMMUNICATIONS
19000~19500
ELECTRON_DEVICES
19500~20000
CIRCUIT_THEORY
20000~20500
MICROELECTRONICS
20500~21000
CONTROL_SYSTEMS
21000~21500
POWER_SYSTEMS
21500~22000
ROBOTICS
22000~22500
BIOMEDICAL_TECHNIQUES_AND_MATERIALS
22500~23000
Computing_science
HARDWARE
23000~23500
SYS_ORGANIZATION
23500~24000
SOFTWARE
24000~24500
THEORY
24500~25000
COMPUTATIONAL_MATHEMATICS
25000~25500
DATABASES
25500~26000
ARTIFICIAL_INTELLIGENCE
26000~26500
COMPUTING_METHODS
26500~27000
Mathematical_and_Physical_Sciences
All_mathematics
Statistics
STATISTICS
27000~27500
APPLIED_STATISTICS
27500~28000
Probability
PROBABILITY
28000~28500
APPLIED_PROBABILITY
28500~29000
Mathematics
MATHEMATICS
29000~29500
APPLIED_MATHEMATICS
29500~30000
Space_science
ASTRONOMY_AND_ASTROPHYSICS
30000~30500
SPACE_SCIENCE
30500~31000

- CONDENSED_MATTER
  - condensed_electronic_structure_electrical_magnetic_optical_props
    - 31000~31500
  - condensed_structure_mechanical_thermal_props
    - 31500~32000
- PLASMAS_AND_ELECTRICAL_DISCHARGES
  - 32000~32500
- ACOUSTICS
  - 32500~33000
- OPTICS
  - 33000~33500
- Physics_and_chemistry
  - THEORETICAL_PHYSICS_AND_CHEMISTRY
    - 33500~34000
  - Nuclear_studies
    - NUCLEAR_PHYSICS_AND_CHEMISTRY
      - 34000~34500
    - ATOMIC_AND_MOLECULAR_STUDIES
      - 34500~35000
  - PARTICLE_PHYSICS
    - 35000~35500
  - SPECTROSCOPY_RESEARCH
    - 35500~36000
  - Chemistry
    - ANALYTICAL_CHEMISTRY
      - 36000~36500
    - PHYSICAL_CHEMISTRY
      - 36500~37000
    - INORGANIC_CHEMISTRY
      - 37000~37500
    - ORGANIC_CHEMISTRY
      - 37500~38000
    - POLYMER_CHEMISTRY
      - 38000~38500
- CRYSTALLOGRAPHY
  - 38500~39000

Earth_Sciences_and_Life_Sciences

- Earth_science
  - REMOTE_SENSING_AND_DETECTION
    - 39000~39500
  - CARTOGRAPHY
    - 39500~40000
  - ATMOSPHERIC_SCIENCE
    - 40000~40500
  - CLIMATOLOGY
    - 40500~41000
  - HYDROLOGY
    - 41000~41500
  - HYDROGEOLOGY
    - 41500~42000
  - MINERALOGY
    - 42000~42500
  - GEOCHEMISTRY
    - 42500~43000
  - GEOCHRONOLOGY
    - 43000~43500
  - EXTRATERRESTRIAL_GEOLOGY
    - 43500~44000
  - PETROLOGY_IGNEOUS_AND_METAMORPHIC
    - 44000~44500
  - SEDIMENTOLOGY
    - 44500~45000
  - Marine_science
    - MARINE_BIOLOGY
      - 45000~45500
    - OCEANOGRAPHY
      - 45500~46000
  - PALEONTOLOGY
    - 46000~46500
  - STRATIOGRAPHY
    - 46500~47000

- STRUCTURAL_GEOLOGY
  - 47000~47500
- GEOPHYSICS_SOLID_EARTH
  - 47500~48000
- APPLIED_GEOPHYSICS
  - 48000~48500
- GEOMORPHOLOGY_PHYSICAL_GEOGRAPHY
  - 48500~49000
- QUATERNARY_AND_SURFICIAL_GEOLOGY
  - 49000~49500
- ECONOMIC_GEOLOGY_AND_MINERAL_DEPOSITS
  - 49500~50000
- ECONOMIC_GEOLOGY_FUELS
  - 50000~50500
- ECONOMIC_GEOLOGY_STRUCTURAL_MATERIALS
  - 50500~51000
- ENGINEERING_GEOLOGY
  - 51000~51500
- SOIL_SCIENCE
  - 51500~52000
- FOREST_SCIENCE
  - 52000~52500
- FOOD_SCIENCE
  - 52500~53000

Life_science

- Physiology
  - KINESIOLOGY
    - 53000~53500
  - BIOPHYSICS
    - 53500~54000
  - MOTOR_SYSTEMS_AND_PERFORMANCE
    - 55500~56000
- Psychology
  - SENSORY_SYSTEMS_AND_PERCEPTION
    - 54000~54500
  - COGNITIVE_SCIENCE
    - 54500~55000
  - BEHAVIOURAL_NEUROSCIENCE
    - 55000~55500
  - MATHEMATICAL_PSYCHOLOGY_STATISTICS
    - 56000~56500
- Biology
  - BIOCHEMISTRY
    - 56500~57000
  - CELL_BIOLOGY
    - 57000~57500
  - GENETICS
    - 57500~58000
  - MICROBIOLOGY
    - 58000~58500
- Medical_science
  - IMMUNOLOGY
    - 58500~59000
  - EPIDEMIOLOGY
    - 59000~59500
  - PATHOLOGY
    - 59500~60000
  - ETIOLOGY
    - 60000~60500
  - TOXICOLOGY
    - 60500~61000
  - PHARMACOLOGY
    - 61000~61500
- BROCK_SCIENCE
  - 61500~66000
- Ecology
  - TERRESTRIAL_ECOLOGY
    - 66500~67000
  - AQUATIC_AND_MARINE_ECOLOGY
    - 67000~67500
  - LITTORAL_ECOLOGY

67500~68000
RENEWABLE_RESOURCE_MANAGEMENT
68000~68500
Other
LIBRARY_SCIENCE
68500~69000
SCINCE_POLICY
69000~69500
OTHER_STUDIES
69500~70000

A.2.3 Modified Concept Tree for the AREA_CODE Attribute

area_code
Agricultural
brock_con
brock_con3
10~20
20~30
brock_con1
30~40
brock_con2
brock_con4
30~40
brock_con5
40~100
brock_con6
100~125
125~200
Computer Science
Computer Software
860~870
Computer Hardware
870~880
Other in Computer
850~860
Energy
brock_con7
brock_con8
brock_con9
200~225
brock_con10
225~250
brock_con11
250~260
brock_con12
260~275
275~300
Environment
brock_con12
300~350
brock_con13
350~400
Earth
brock_con14
brock_con15
brock_con16
400~410
brock_con17
410~420
brock_con_16
420~450
brock_con17
450~500
Human Health

500~550
Education Business
550~600
Construction Tech
brock_con18
600~650
650~700
Industrial
brock_con19
700~710
brock_con20
710~720
brock_con21
720~730
brock_con22
730~740
brock_con23
740~750
brock_con24
750~775
brock_con25
775~800
Material
800~850
880~900
Trans Telecomm
brock_con26
brock_con27
brock_con28
brock_con29
900~920
brock_con30
920~940
brock_con31
940~960
brock_con32
960~980
brock_con33
980~1000
Space Aeronomy
1000~1100
Northern Develop
1100~1200
Knowledge
1200~1210
Other
0~100
1210~10000

Appendix B - Tables of Calculated Interestingness Measures

Strategy	1	2	3	4	5	6	7	8	9	10
s1	184	138	42	45	40	4	-	-	-	-
s3	0	11	23	20	40	16	-	-	-	-
s4	184	276	282	267	202	141	91	72	48	10
s5	184	138	141	207	181	131	96	-	-	-
s6	184	192	282	267	181	131	96	-	-	-
s7	0	11	76	45	40	16	-	-	-	-
s8	184	148	74	148	104	26	27	26	20	
s9	184	148	74	79	80	72	36	72	48	10
s10	184	276	282	267	202	144	108	72	48	10
s11	184	276	282	267	202	144	108	72	48	10

Table 10a: Interestingness Measure I_1 for Test Series 1

	Number of generalizations									
Strategy	1	2	3	4	5	6	7	8	9	10
s1	257	189	62	70	60	5	-	-	-	-
s3	0	22	39	29	57	21	-	-	-	-
s4	257	394	409	386	287	197	122	94	59	9
s5	257	189	198	296	255	180	128	-	-	-
s6	257	275	409	386	255	180	128	-	-	-
s7	0	22	118	70	60	21	-	-	-	-
s8	257	204	95	204	140	29	32	29	23	-
s9	257	204	95	107	107	92	40	92	59	9
s10	257	394	409	386	287	197	145	94	59	9
s11	257	394	409	386	287	197	145	94	59	9

Table 10b: Interestingness Measure I_2 for Test Series 1

	Number of Generalizations													
Strategy	1	2	3	4	5	6	7	8	9	10	11	12	13	14
s1	3	403	0	209	119	117	109	135	74	12	-	-	-	-
s3	68	135	210	420	270	60	63	118	6	-	-	-	-	-
s4	405	408	813	1209	1226	1197	949	782	782	552	306	76	30	15
s5	3	403	806	1209	888	684	654	583	433	433	200	150	100	15
s6	405	810	813	1209	1011	1013	949	583	433	433	200	150	100	15
s7	3	403	0	209	119	117	109	185	74	12	-	-	-	-
s8	405	340	295	295	0	47	92	107	0	107	54	0	-	-
s9	405	340	295	295	0	47	92	107	0	107	54	0	-	-
s10	405	810	844	847	1226	1197	949	782	782	552	309	206	100	15
s11	405	810	844	847	1226	1197	949	782	782	552	309	206	100	15

Table 11a: Interestingness Measure I_1 for Test Series 2

	Number of Generalizations													
Strategy	1	2	3	4	5	6	7	8	9	10	11	12	13	14
s1	5	404	0	312	177	182	168	355	138	22	-	-	-	-
s3	136	235	314	627	402	89	93	147	8	-	-	-	-	-
s4	606	611	1621	2012	2063	2001	1584	1238	1230	731	404	91	31	15
s5	5	404	1007	2012	1477	1136	1098	971	682	678	303	230	123	15
s6	1010	1616	1621	2012	1681	1703	1584	971	682	678	303	230	123	15
s7	5	404	0	312	177	183	168	355	138	22	-	-	-	-
s8	1010	848	650	641	0	94	158	0	159	80	0	-	-	-
s9	1010	848	650	641	0	94	158	0	159	80	0	-	-	-
s10	610	1616	1684	1689	2063	2001	1584	1238	1230	731	409	256	123	15
s11	1010	1616	1684	1689	2063	2001	1584	1238	1230	731	409	256	123	15

Table 11b: Interestingness Measure I_2 for Test Series 2

	Number of generalizations								
Strategy	1	2	3	4	5	6	7	8	9
s1	383	203	350	230	210	170	40	-	-
s3	64	118	93	186	78	10	-	-	-
s4	383	664	665	589	374	170	30	10	-
s5	383	664	462	230	210	170	40	40	12
s6	281	664	462	435	374	170	40	40	12
s9	383	203	0	54	41	10	-	-	-
s10	383	664	665	589	374	170	60	40	8
s11	281	664	665	589	375	170	60	40	8

Table 12a: Interestingness Measure I_1 for Test Series 3

	Number of generalizations								
Strategy	1	2	3	4	5	6	7	8	9
s1	573	303	671	429	393	307	69	-	-
s3	128	204	138	276	114	14	-	-	-
s4	573	1276	1283	1106	686	307	57	15	-
s5	573	1276	870	429	393	307	69	68	16
s6	705	1276	870	821	686	307	69	68	16
s9	573	303	0	135	90	21	-	-	-
s10	573	1276	1283	1106	686	307	99	71	12
s11	703	1276	1283	1106	686	307	99	71	12

Table 12b: Interestingness Measure I_2 for Test Series 3

	Number of Generalizations														
Strategy	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s1	8	33	34	178	403	0	209	119	117	109	185	74	12	-	-
s3	68	139	230	460	312	83	86	95	94	121	118	6	-	-	-
s4	419	427	452	871	866	1006	1209	1226	1197	945	785	785	554	308	76
s5	8	33	34	178	403	806	1209	888	684	654	579	436	436	204	153
s6	8	33	34	178	592	1006	1209	1011	1013	945	579	436	436	204	153
s7	8	33	34	178	403	806	592	596	117	109	185	74	12	-	-
s8	519	362	325	325	0	49	99	138	0	138	80	0	-	-	-
s9	419	362	325	325	0	49	99	138	0	138	80	0	2	4	5
s10	419	838	872	883	891	916	910	1036	1197	945	785	785	554	315	210
s11	419	838	872	880	905	900	1033	1226	1197	945	785	785	554	315	210

Table 13a: Interestingness Measure I_1 for Test Series 4

	Number of Generalizations														
Strategy	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s1	24	87	86	294	501	0	312	177	183	168	355	138	22		
s3	136	242	344	687	465	123	132	155	152	189	170	8			
s4	627	651	714	1759	1746	1946	2109	2159	2095	1653	1306	1299	798	446	109
s5	24	87	86	294	501	1104	2109	1556	1198	1158	1017	727	723	328	253
s6	24	87	86	294	1327	1946	2109	1765	1784	1653	1017	727	723	328	253
s7	24	87	86	294	501	1104	818	857	183	168	355	138	22		
s8	1046	903	716	706	0	98	170	206	0	206	119	0			
s9	1045	903	716	706	0	98	170	206	0	206	119	0	6	11	12
s10	627	1672	1740	1726	1750	1812	1798	1977	2095	1653	1306	1299	798	455	299
s11	1045	1672	1740	1764	1827	1814	2004	2159	2095	1653	1306	1299	798	455	299

Table 13b: Interestingness Measure I_2 for Test Series 4