INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

.

.

**ATTRIBUTE SELECTION STRATEGIES FOR
ATTRIBUTE-ORIENTED GENERALIZATION**

**A Thesis
Submitted to the Faculty of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the Degree of
Master of Science
in
Computer Science
University of Regina**

**By
David Brock Barber
Regina, Saskatchewan
March, 1997**

© Copyright D. Brock Barber, 1997, All Rights Reserved

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-30442-6

Canada

UNIVERSITY OF REGINA
FACULTY OF GRADUATE STUDIES AND RESEARCH
PERMISSION TO USE POSTGRADUATE THESES

TITLE OF THESIS: *Attribute Selection Strategies for Attribute-Oriented Generalization*

NAME OF AUTHOR: David Brock Barber

FACULTY: Faculty of Graduate Studies and Research

DEGREE: Master of Science

In presenting this thesis in partial fulfillment of the requirements for a postgraduate degree from the University of Regina, I agree that the Libraries of this University shall make it freely available for inspection. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the professor or professors who supervised my thesis work, or in their absence, by the Head of the Department or the Dean of the Faculty in which my thesis work was done. It is understood that any copying, publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Regina in any scholarly use which may be made of my material in my thesis.

SIGNATURE: Brock Barber

DATE: March 21, 1997

UNIVERSITY OF REGINA

FACULTY OF GRADUATE STUDIES AND RESEARCH

CERTIFICATION OF THESIS WORK

We, the undersigned, certify that David Brock Barber, candidate for the Degree of Master of Science, has presented a thesis on *Attribute Selection Strategies for Attribute-Oriented Generalization*, that the thesis is acceptable in form and content, and that the student demonstrated a satisfactory knowledge of the field covered by the thesis in an oral examination held on March 21, 1997.

External Examiner:

Dr. J. Aggarwal, Faculty of Administration

Internal Examiners:

Dr. H.J. Hamilton, Supervisor

(S.E. AHMED)

Abstract

Attribute-oriented generalization (AOG) is a knowledge discovery method that summarizes and simplifies the information in a relational database by repeatedly replacing specific attribute values with more general concepts. Several strategies for selecting the next attribute to generalize have been suggested in the literature but their relative merits have not previously been assessed. This thesis describes and compares these strategies, as well as several new strategies.

The performance of the strategies is evaluated and compared using criteria that consider their ability to efficiently produce interesting results. Measures of interestingness are used that consider the structure of the domain-expert defined concept hierarchies used to guide the generalization process. Based on the experimental results, a strategy that selects the attribute for generalization that has the greatest potential for contributing interesting concepts to subsequent generalized relations was found to be the most efficient and effective of the strategies tested.

Acknowledgments

I wish to thank my wife Diane for her unquestioning love and support in this project and in all other things. I'd like to express my thanks to Dr. Howard Hamilton for his guidance, assistance and financial support during the completion of this thesis. He is a rare combination of excellent researcher, supervisor, educator and most importantly, a fine human being. Additional thanks go to the Faculty of Graduate Studies and Research and Sasktel for their financial support, the members of my committee and external examiner for their valuable input and to the other faculty members, fellow graduate students and friends who have made the learning experience enjoyable.

Table of Contents

Acknowledgments	i
Abstract	ii
Table of Contents	iii
List of Figures	vi
List of Tables	viii
CHAPTER 1 - INTRODUCTION	1
1.1 Knowledge Discovery in Databases (KDD)	1
1.2 Attribute-oriented Generalization (AOG)	2
1.3 Thesis Organization	3
CHAPTER 2 - KNOWLEDGE DISCOVERY IN DATABASES	4
2.1 Introduction	4
2.2 Characteristics of Database Data	6
2.3 The Role of Domain Knowledge in KDD	9
2.4 Interestingness of Discovery	10
2.4.1 Introduction	10
2.4.2 An Overview of Interestingness Measures Used in KDD	11
2.5 Inductive Learning and Knowledge Discovery Algorithms	17
2.6 Knowledge Representation	19
CHAPTER 3 - ATTRIBUTE-ORIENTED GENERALIZATION	22
3.1 Introduction	22
3.2 Description of Attribute-oriented Generalization	23
3.3 Example of Attribute-oriented Generalization	27
3.4 Domain Knowledge	29
3.5 Interestingness of Results	31
3.6 Strengths and Weaknesses of AOG	31

CHAPTER 4 - ELIMINATION OF THE ATTRIBUTE THRESHOLD	33
4.1 Introduction.....	33
4.2 Criteria for the Evaluation of Attribute Selection Strategies	35
4.2.1 Introduction	35
4.2.2 Interestingness Criteria	36
4.3 Attribute Selection Strategies	40
4.3.1 Introduction.....	40
4.3.2 Attribute Selection Strategy Classification	42
4.3.3 Lookahead Strategies	43
4.3.3.1 Selection Based on the Number of Tuples after Generalization (S2 and S3).....	43
4.3.3.2 Selection Based on Interestingness Measures I1 and I2 (S9 and S10)	44
4.3.4 Predictive Strategies.....	46
4.3.4.1 Selection based on the Number of Distinct Attribute Values (S1 and S8).....	46
4.3.4.2 Strategies Based on the Complexity of the Hierarchy Trees (S4 and S5).....	47
4.4 Time Complexity Analysis	49
CHAPTER 5 - ANALYSIS AND DISCUSSION OF RESULTS	53
5.1 Introduction.....	53
5.2 The DB-Discover Program	53
5.3 Description of Test Databases.....	55
5.4 Test Program and Experimental Procedure	56
5.5 Method of Evaluation	57
5.6 Analysis of the Lookahead Strategies	61
5.6.1 Analysis of Strategy S2	61
5.6.2 Analysis of Strategy S3	64
5.6.3 Analysis of Strategies S9 and S10	67
5.7 Analysis of Predictive Strategies	72
5.7.1 Analysis of Strategy S1	72
5.7.2 Analysis of Strategy S8	74
5.7.3 Analysis of Strategies S4 and S5	76
5.8 Comparison of the Attribute Selection Strategies	80
5.9 Analysis of Modified Strategies S11 and S12	84
CHAPTER 6 - CONCLUSIONS	91
6.1 Attribute Selection for Generalization in AOG	91
6.2 Areas for Future Research	93

REFERENCES.....	95
APPENDIX A - CHARTS OF $%d_{2a}$ VERSUS TUPLE RANGE	99
APPENDIX B - STATISTICAL ANALYSIS OF	
EXPERIMENTAL RESULTS	104
A.1 Summary of Means and Standard deviations for $%d_{1a}$ and $%d_{2a}$ values	104
A.2 Summaries of the Statistical Comparisons of $%d_{1a}$ and $%d_{2a}$ values	112

List of Figures

Figure 2.1 KDD System Model	6
Figure 3.1 Concept Hierarchy for the Province Attribute	24
Figure 3.2 Concept Hierarchy for the Grant_Amount Attribute	24
Figure 3.3 Illustration of the Generalization Process	28
Figure 4.1 Attribute Generalization Graph	34
Figure 4.2 Concept Hierarchies for Sample Relation Attributes	41
Figure 5.1 I_l and I_2 versus Number of Tuples for all Strategies in Task T27	59
Figure 5.2 Example Chart of $\%d_{la}$ versus Tuple Range	60
Figure 5.3 Strategy S2 I_l versus Number of Tuples for Task T27	62
Figure 5.4 $\%d_{la}$ versus Tuple Range for Strategy S2	63
Figure 5.5 Strategy S3 I_l versus Number of Tuples for Task T27	65
Figure 5.6 $\%d_{la}$ versus Tuple Range for Strategy S3	67
Figure 5.7 Strategies S9 and S10 I_l versus Number of Tuples for Task T27	67
Figure 5.8 $\%d_{la}$ versus Tuple Range for Strategy S9	71
Figure 5.9 $\%d_{la}$ versus Tuple Range for Strategy S10	71
Figure 5.10 Strategy S1 I_l versus Number of Tuples for Task T27	72
Figure 5.11 $\%d_{la}$ versus Tuple Range for Strategy S1	73
Figure 5.12 Strategy S8 I_l versus Number of Tuples for Task T27	74
Figure 5.13 $\%d_{la}$ versus Tuple Range for Strategy S8	76

Figure 5.14 Strategies S4 and S5 I_l versus Number of Tuples for Task T27	77
Figure 5.15 $\%d_{1a}$ versus Tuple Range for Strategy S4	79
Figure 5.16 $\%d_{1a}$ versus Tuple Range for Strategy S5	80
Figure 5.17 Strategies S11 and S12 I_l versus Number of Tuples for Task T27	85
Figure 5.18 $\%d_{1a}$ versus Tuple Range for Strategy S11	87
Figure 5.19 $\%d_{1a}$ versus Tuple Range for Strategy S12	87
Figure A.1 $\%d_{2a}$ versus Tuple Range for Strategy S1.....	99
Figure A.2 $\%d_{2a}$ versus Tuple Range for Strategy S2.....	99
Figure A.3 $\%d_{2a}$ versus Tuple Range for Strategy S3.....	100
Figure A.4 $\%d_{2a}$ versus Tuple Range for Strategy S4.....	100
Figure A.5 $\%d_{2a}$ versus Tuple Range for Strategy S5.....	101
Figure A.6 $\%d_{2a}$ versus Tuple Range for Strategy S8.....	101
Figure A.7 $\%d_{2a}$ versus Tuple Range for Strategy S9.....	102
Figure A.8 $\%d_{2a}$ versus Tuple Range for Strategy S10.....	102
Figure A.9 $\%d_{2a}$ versus Tuple Range for Strategy S11.....	103
Figure A.10 $\%d_{2a}$ versus Tuple Range for Strategy S12.....	103

List of Tables

Table 3.1 Sample Target Relation.....	28
Table 3.2 Relation After Generalization to Attribute and Table Thresholds.....	29
Table 4.1 Sample Relation.....	41
Table 4.2 Relation R_A	42
Table 4.3 Relation R_B	42
Table 4.4 Attribute Selection Strategy Classification.....	43
Table 4.5 Values for Determination of Interestingness Measures	45
Table 4.6 Values for Determination of Complexity Measures	49
Table 4.7 Summary of Time Complexity Analysis for a Single Generalization	52
Table 5.1 Sample Test Results for One Attribute Selection Strategy.....	57
Table 5.2 Strategy S2 Generalization Information for Task T27.....	62
Table 5.3 Strategy S3 Generalization Information for Task T27.....	65
Table 5.4 Strategy S9 Generalization Information for Task T27.....	69
Table 5.5 Strategy S10 Generalization Information for Task T27.....	69
Table 5.6 Strategy S1 Generalization Information for Task T27.....	73
Table 5.7 Strategy S8 Generalization Information for Task T27.....	75
Table 5.8 Strategy S4 Generalization Information for Task T27.....	77
Table 5.9 Strategy S5 Generalization Information for Task T27.....	78
Table 5.10 Strategies Ranked by $\%d_{la}$ for SaskTel Tasks.....	81

Table 5.11 Strategies Ranked by $\%d_{1a}$ for NSERC Tasks.....	81
Table 5.12 Strategies Ranked by $\%d_{2a}$ for SaskTel Tasks.....	82
Table 5.13 Strategies Ranked by $\%d_{2a}$ for NSERC Tasks.....	82
Table 5.14 Strategy S11 Generalization Information for Task T27.....	86
Table 5.15 Strategy S12 Generalization Information for Task T27.....	86
Table 5.16 Comparison of $\%d_{1a}$ of Modified Strategies for SaskTel Tasks.....	88
Table 5.17 Comparison of $\%d_{1a}$ of Modified Strategies for NSERC Tasks.....	89
Table 5.18 Comparison of $\%d_{2a}$ of Modified Strategies for SaskTel Tasks.....	89
Table 5.19 Comparison of $\%d_{2a}$ of Modified Strategies for NSERC Tasks.....	89
Table B.1: Tuple Range 1-10 $\%d_{1a}$ and $\%d_{2a}$ Values for SaskTel Tasks	105
Table B.2: Tuple Range 11-20 $\%d_{1a}$ and $\%d_{2a}$ Values for SaskTel Tasks	106
Table B.3: Tuple Range 21-30 $\%d_{1a}$ and $\%d_{2a}$ Values for SaskTel Tasks	106
Table B.4: Tuple Range 31-40 $\%d_{1a}$ and $\%d_{2a}$ Values for SaskTel Tasks	106
Table B.5: Tuple Range 41-50 $\%d_{1a}$ and $\%d_{2a}$ Values for SaskTel Tasks	107
Table B.6: Tuple Range 51-60 $\%d_{1a}$ and $\%d_{2a}$ Values for SaskTel Tasks	107
Table B.7: Tuple Range 61-70 $\%d_{1a}$ and $\%d_{2a}$ Values for SaskTel Tasks	107
Table B.8: Tuple Range 71-80 $\%d_{1a}$ and $\%d_{2a}$ Values for SaskTel Tasks	108
Table B.9: Tuple Range 81-90 $\%d_{1a}$ and $\%d_{2a}$ Values for SaskTel Tasks	108
Table B.10: Tuple Range 91-100 $\%d_{1a}$ and $\%d_{2a}$ Values for SaskTel Tasks	108
Table B.11: Tuple Range 1-10 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks	109
Table B.12: Tuple Range 11-20 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks	109
Table B.13: Tuple Range 21-30 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks	109

Table B.14: Tuple Range 31-40 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks	110
Table B.15: Tuple Range 41-50 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks	110
Table B.16: Tuple Range 51-60 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks	110
Table B.17: Tuple Range 61-70 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks	111
Table B.18: Tuple Range 71-80 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks	111
Table B.19: Tuple Range 81-90 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks	111
Table B.20: Tuple Range 91-100 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks	112
Table B.21: Tuple Range 1-10 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	114
Table B.22: Tuple Range 11-20 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	115
Table B.23: Tuple Range 21-30 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	115
Table B.24: Tuple Range 31-40 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	115
Table B.25: Tuple Range 41-50 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	116
Table B.26: Tuple Range 51-60 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	116
Table B.27: Tuple Range 61-70 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	116
Table B.28: Tuple Range 71-80 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	117
Table B.29: Tuple Range 81-90 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	117
Table B.30: Tuple Range 91-100 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	117

Table B.31: Tuple Range 1-10 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	118
Table B.32: Tuple Range 11-20 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	118
Table B.33: Tuple Range 21-30 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	118
Table B.34: Tuple Range 31-40 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	119
Table B.35: Tuple Range 41-50 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	119
Table B.36: Tuple Range 51-60 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	119
Table B.37: Tuple Range 61-70 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	120
Table B.38: Tuple Range 71-80 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	120
Table B.39: Tuple Range 81-90 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	120
Table B.40: Tuple Range 91-100 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	121
Table B.41: Tuple Range 1-10 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	121
Table B.42: Tuple Range 11-20 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	121
Table B.43: Tuple Range 21-30 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	122
Table B.44: Tuple Range 31-40 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	122
Table B.45: Tuple Range 41-50 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	122

Table B.46: Tuple Range 51-60 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	123
Table B.47: Tuple Range 61-70 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	123
Table B.48: Tuple Range 71-80 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	123
Table B.49: Tuple Range 81-90 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	124
Table B.50: Tuple Range 91-100 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks	124
Table B.51: Tuple Range 1-10 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	124
Table B.52: Tuple Range 11-20 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	125
Table B.53: Tuple Range 21-30 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	125
Table B.54: Tuple Range 31-40 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	125
Table B.55: Tuple Range 41-50 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	126
Table B.56: Tuple Range 51-60 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	126
Table B.57: Tuple Range 61-70 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	126
Table B.58: Tuple Range 71-80 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	127
Table B.59: Tuple Range 81-90 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	127
Table B.60: Tuple Range 91-100 z Values for $\%d_{1a}$ Comparison for NSERC Tasks	127

CHAPTER 1

INTRODUCTION

1.1 Knowledge Discovery in Databases (KDD)

Knowledge discovery in databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad, Piatetski-Shapiro and Smyth, 1996). The development of practical systems for automated discovery of knowledge from data has become increasingly important as scientific, government and business databases have grown, both in size and complexity. Exhaustive, empirical analysis is all but impossible on the megabytes, gigabytes or even terabytes of data in real-world databases (Matheus, Chan and Piatetsky-Shapiro, 1993).

A system for discovering knowledge in databases typically consists of a knowledge discovery algorithm whose input is data from a database and domain knowledge and whose output is discovered knowledge in a form that is understandable to the user. When dealing with potentially huge databases, it is a practical necessity to use domain knowledge in a knowledge discovery system (Anand, Bell and Hughes, 1995). Domain knowledge consists of domain specific information that constrains the search space and guides the system towards interesting results.

A number of successful methods for performing knowledge discovery in databases have been developed. One of these methods is attribute-oriented generalization, which is introduced in the following section.

1.2 Attribute-oriented Generalization (AOG)

Attribute-oriented generalization is a knowledge discovery method that uses generalization to simplify the descriptions of patterns in data. Domain knowledge, in the form of hierarchical classifications of concepts, guides the generalization of each attribute in a relation retrieved from a database. Specific attribute values in a relation are repeatedly replaced with more general concepts. As related specific attribute values are grouped together into more general concepts, 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 degree of generalization is governed by two thresholds. An *attribute threshold* T_a for each attribute specifies 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 maximum number of tuples permitted in the final generalized relation.

As presented by other researchers, the AOG process does not consider how interesting the results will be to the user. It is difficult to choose values of the user-specified thresholds that will produce useful and interesting information. The goal of the research described in this thesis is to replace the attribute threshold with a strategy for selecting attributes for generalization that will guide the user towards the most interesting results. This thesis describes an experimental investigation of different strategies for selecting the next attribute to generalize. The strategies are evaluated according to their ability to produce new and interesting results and their efficiency in producing these results. Other than an initial report on the results of this research (Barber and Hamilton, 1996), no previous comparison of these strategies has appeared in the literature.

1.3 Thesis Organization

Chapter 2 presents a simple model of a knowledge discovery system and explores important aspects of the system. The characteristics of database data that must be considered in the design of a knowledge discovery system are outlined. The use of domain knowledge in knowledge discovery systems is discussed. The specification of what constitutes interesting discovery is examined and an overview of some methods for evaluating interestingness that have been applied in KDD is provided. Inductive learning and some common knowledge discovery algorithms are described. The representation of discovered knowledge is addressed. Chapter 3 provides a general description of attribute-oriented generalization and an example to illustrate its use. The use of domain knowledge in AOG is discussed. Some strengths and weakness of AOG are identified. Chapter 4 examines the replacement of the attribute threshold with a strategy for selecting the next attribute to generalize. Criteria for evaluating attribute selection strategies are detailed. Attribute selection strategies suggested in the literature are discussed and new strategies are presented. Examples illustrate the application of the various strategies to a sample relation. Chapter 5 introduces the DB-Discover knowledge discovery program, which uses AOG to discover knowledge from databases, and explains the experimental approach that was used to test the attribute selection strategies. The experimental results are presented and discussed. Chapter 6 presents the conclusions of the experimental study and identifies potential areas of future research.

CHAPTER 2

KNOWLEDGE DISCOVERY

IN DATABASES

2.1 Introduction

Increasingly large and complex databases are being accumulated by business, government and scientific organizations. The data contained in these databases are a potentially valuable resource for an organization. However, the volume of data often greatly exceeds the ability to analyze it, which has led to a growing gap between data generation and data understanding (Frawley, Piatetsky-Shapiro and Matheus, 1991). Database query systems and statistical packages are not sufficient to close the gap and intelligent support systems for discovery of interesting findings in databases are becoming increasingly important (Klösgen, 1992). Knowledge discovery in databases (KDD) describes a field that uses concepts and techniques from areas such as artificial intelligence, statistics and database search and management to discover knowledge from the data in databases.

Recall that *knowledge discovery* is the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data. A system for discovering knowledge in data from a database should efficiently extract patterns or regularities and present them in an understandable manner. The presented patterns should be interesting and certain enough to be useful.

The primary foundation of KDD is provided by the field of machine learning. However, the data found in real databases have different characteristics than the data sets typically used in machine learning applications. Because of this, many machine learning algorithms are rendered ineffective in the general case when applied to real databases (Frawley, Piatetsky-Shapiro and Matheus, 1991). KDD algorithms are typically modified machine learning algorithms that consider the characteristics of database data.

Figure 2.1 shows a simple model of a system for discovering knowledge in databases. The model is an adaptation of the framework for knowledge discovery in databases described by Frawley, Piatetsky-Shapiro and Matheus (1991). The discovery method is the algorithm used for knowledge discovery. The input to the discovery method is data from the database and domain knowledge. Domain knowledge can be incorporated into the system, obtained interactively from the user or consist of knowledge discovered by the system itself. The output from the discovery method is discovered knowledge represented in a form that is understandable and useful to the user.

Section 2.2 describes typical properties of database data that must be considered in KDD applications. Section 2.3 provides a discussion of the role of domain knowledge in knowledge discovery. Section 2.4 addresses the problem of identifying interesting discoveries and briefly describes methods for evaluating interestingness that have been applied in KDD. Section 2.5 discusses common inductive learning strategies used in KDD. Finally, Section 2.6 describes output knowledge representation.

Figure 2.1: KDD System Model

2.2 Characteristics of Database Data

KDD has strong roots in the field of machine learning. However, as mentioned, some characteristics of real world databases are not encountered in the data used in most traditional machine learning applications. These characteristics present unique problems in the task of knowledge discovery.

The volume of data that is stored in a database can be huge, potentially amounting to terabytes of data (Frawley, Piatetsky-Shapiro and Matheus, 1991), (Fayyad, Wier and Djorgovski, 1993). The three terabyte database explored by the SKYCAT system (Fayyad, Wier and Djorgovski, 1996) is expected to contain information about 2 billion

objects. The database examined by the RADIX/RX system contains objects with 40 to 40,000 attributes (Holsheimer and Siebes, 1994). Given the potential magnitude of the data set on which the discovery must be performed, the efficiency of a knowledge discovery system (KDS) is an important consideration. Even with efficient algorithms, the sheer size of the database often necessitates the use of sampling to reduce the magnitude of the data set and the use of domain knowledge to guide and constrain the discovery process.

The data in databases are sparse because most combinations of data values are not present in the data. Attributes needed for accurate learning may not be present in the database. This leads to uncertainty in the validity of discovered knowledge. Other factors affecting uncertainty include the sample size, the integrity of the data, the degree of support from existing domain knowledge and the inherent indeterminism of the underlying real-world causes (Frawley, Piatetsky-Shapiro and Matheus, 1991).

The data in a database are dynamic. Data values are added, deleted or modified. Discovery results can become outdated. The change in data values is often interesting to the user, since it can represent variations in the behavior of important processes or trends in the environment. A KDS should be able to adapt to changes in the data. A flexible knowledge discovery technique should be performed incrementally on database updates to avoid restarting the costly learning process from the beginning (Han, Cai and Cercone, 1993).

Typically, a database is created and maintained for some purpose other than knowledge discovery. The database may contain attributes that are not relevant to a

discovery task and it may not contain attributes that are relevant. Although the data in a database are dynamic, the structure of the database is relatively static and cannot easily be changed to facilitate discovery. In addition, if the database contains historical information, it may be impossible to obtain additional historical data needed for the discovery process.

Database data tends to be noisy and values for attributes are often missing. If the data in the database are collected and entered by humans, it typically contains errors. Attributes may be based on measurements or subjective judgments, which may be subject to errors. Non-systematic errors of this kind in the values of attributes are usually referred to as noise (Holsheimer and Siebes, 1994). This noise can affect the results of a discovery task.

A database typically contains positive examples but lacks explicit negative examples. Positive examples provide instances of the concepts to be acquired but do not provide information to prevent overgeneralization of inferred concepts (Carbonell, Michalski and Mitchell, 1983). To avoid overgeneralization, it may be necessary to perform minimal generalizations or to use domain knowledge to constrain the concept generalization.

Despite the additional difficulties involved in performing discovery on data in a real world database, it is possible to use modified machine learning algorithms for the purposes of KDD. Typically, the modified algorithms use domain knowledge to guide and constrain discovery or perform the discovery on subsets of the database.

2.3 The Role of Domain Knowledge in KDD

A traditional goal in machine learning has been to develop a machine that can acquire all knowledge starting with nothing. Recently, there has been a paradigm shift in machine learning towards the gradual realization that a machine can do much better in learning tasks with a little appropriate prior knowledge (Smyth and Goodman, 1991). From a practical standpoint, background knowledge is necessary to reduce an infinite number of possible hypothesis that can be generated to imply a given set of facts to a few preferred ones (Michalski, 1983). For this reason, machine learning and KDD systems usually utilize domain knowledge.

Domain knowledge is domain specific information, usually provided by a domain expert, that is used to guide and constrain the search for interesting information. Domain knowledge is usually incorporated into the discovery system or obtained interactively during system execution, although it can potentially be discovered knowledge. In KDD, domain knowledge can provide information about the form and content of data, the domain(s) described by the database, and the context and purpose of the current discovery effort. This might include the relevance of attributes to a particular discovery task, relationships between attributes, data dictionary information or the representation of discovered results that would make them more understandable to the user. The primary disadvantage of using domain knowledge in the discovery process is that it introduces user bias. If the system is told what to look for, it is unlikely to discovery useful, unexpected results.

One area where domain knowledge plays a key role in many KDD systems is the specification of what constitutes an interesting discovery. This aspect of domain knowledge is discussed in greater detail in the following section.

2.4 Interestingness of Discoveries

2.4.1 Introduction

One problem in designing a knowledge discovery algorithm is the specification of an adequate method of determining the interestingness of discovered knowledge. Many patterns are contained in the data for any database, but few of these are likely to be interesting to the user. If the user is presented with many patterns, most of which are of no interest, the truly interesting patterns may be obscured. Clearly, it is desirable to present only the interesting patterns to the user.

The concept of interestingness can be used by a knowledge discovery system in two ways (Silberschatz and Tuzhilin, 1995). Interestingness can be used as a filter attached at the back end of a knowledge discovery system. The filter is used to select the interesting patterns from among the many patterns discovered by the system. One advantage of this method is that the filter can be easily changed to accommodate a shift in interest. Alternatively, the concept of interestingness can be incorporated as an integral part of the knowledge discovery system. Interestingness can be used to guide the discovery process so that the knowledge discovery system finds only the interesting patterns. The advantage of interestingness guided discovery is that most uninteresting patterns are not generated and the search space is reduced.

The interestingness of a discovery is often dependent on the context of the current discovery effort and user defined bias. Silberschatz and Tuzhilin (1995) classify interestingness measures as objective or subjective measures. An *objective interestingness measure* is one that depends only on the structure of the pattern and the underlying data. A *subjective interestingness measure* also considers what the user considers to be interesting.

Frawley, Piatetsky-Shapiro and Matheus (1991) define an interesting pattern as one that is novel, useful and non-trivial to compute. This definition is sufficiently general to cover a wide spectrum of potentially viable interestingness methods, some of which are described in the following section.

2.4.2 An Overview of Interestingness Measures Used in KDD

Piatetsky-Shapiro (1991) uses a *rule-interest function* RI to prune uninteresting rules represented in the form $A \rightarrow B$. The function is based on a comparison of the actual number of tuples in which conditions A and B are satisfied and the expected number of tuples if A and B were statistically independent. A positive value for RI indicates a positive correlation between the conditions, while a negative value indicates a negative correlation. The statistical significance of the correlation is determined and a user-specified minimum significance threshold is used to prune uninteresting rules. User-specified thresholds are also used to remove rules that are not supported by enough instances in the data and rules that are too general.

Bhandari (1993) suggests two interestingness functions for ordering all possible subsets of an input data set that can be derived by specifying a set of attributes and their values. The first function, referred to as I_1 , ranks potential output by measuring the difference between the number of records containing a specified attribute value and the number expected if the values for that attribute were uniformly distributed.

The second function, referred to as I_2 , ranks potential output by measuring the difference between the proportion of records that have a specified pair of attribute values and the proportion one would expect if the values were statistically independent. The function I_2 is the same as the rule-interest function used by Piatetsky-Shapiro (1991). However, the magnitude of the difference signifies interestingness, rather than a statistical measure of the significance of the difference. Using the significance of the difference rather than the magnitude of the difference allows the pruning of differences that could occur by chance.

Once the subsets are ordered, a filtering function is used to determine which should be presented to the user. The filtering function uses the ordering of the subsets by interestingness measure and parameters for the limits of human understanding to determine the number of allowed tables and the number of related items allowed in these tables.

Feldman and Dagan (1995) use the concept of relative entropy from information theory to measure interestingness of concept distributions in the text files. The text files are annotated with concepts from user-defined concept hierarchies. If the actual distribution of concepts is $p(x)$ and the expected distribution of the concepts is $q(x)$, then

the *relative entropy* or *Kullback-Leibler distance* between $p(x)$ and $q(x)$ is a measure of the surprise in seeing the distribution $p(x)$. A large relative entropy measure indicates an interesting pattern. A concept that has a large influence on an interesting relative entropy measure is also considered to be interesting.

Smyth and Goodman (1991) use a J-measure to express the quality of discovered probabilistic rules in the form if $Y = y$ then $X = x$ with probability p . The *J-measure* is the average information content of a rule (Smyth and Goodman, 1992), calculated as the product of the probability of event y and an entropy measure called the *j-measure*. The *j-measure* is a specialized case of relative or cross entropy that indicates the average mutual information between events x and y with the expectation taken with respect to the a posteriori probability distribution of X . The intent of the J-measure is to provide a balance between the simplicity of a rule and its information content.

Matheus and Piatetsky-Shapiro (1995) use deviations to identify potentially interesting patterns. *Deviations* are defined as differences between measured values and corresponding references such as previous values or normative values. A significant deviation in a measure is potentially interesting. However, the actual interestingness of a deviation is determined by the estimated benefit that would occur from an available action in response to the deviation. The *estimated benefit* is the future financial saving that would result from taking an action to reduce the deviation. The *projected saving* is determined as a product of the projected impact of a deviation and saving percentage. The *projected impact* is the projected deviation from the norm translated into a projected financial impact on the bottom line. The *saving percentage* is a domain expert specified

value of the percentage decrease in a deviation that would occur as the result of some action. A minimum threshold is used to eliminate deviations lacking adequate instance support. Heuristic rules are used to consider extreme deviations based on a small number of instances.

Silberschatz and Tuzhilin (1991) consider a pattern as evidence that can potentially affect a system of hard and soft beliefs. A *hard belief* represents a constraint that cannot be changed with new evidence. Therefore, any evidence that does contradict a hard belief is always considered to be interesting. Such contradictory evidence indicates that the data underlying a pattern is incorrect and should be changed. A *soft belief* is one that the user is willing to change as new patterns are discovered that provide new evidence. Each soft belief has an associated degree of belief indicating the amount of confidence in the belief. Supporting evidence increases the degree of belief and contradictory evidence decreases the degree of belief. The more that a new pattern changes the confidence in a set of soft beliefs, the more interesting it is.

Hamilton and Fudger (1995) use two interestingness measures, referred to as I_1 and I_2 , to evaluate the results from a system that uses attribute-oriented generalization to discover interesting information from a database. The generalization is guided by domain-expert defined concept hierarchies for the attributes. An interesting attribute value in an output relation is one corresponding non-leaf, non-root concepts in the appropriate concept hierarchies. The interestingness measure I_1 is simply a count of the interesting values in a relation. The second interestingness measure I_2 considers the

structure of the concept hierarchy in assigning interestingness values to concepts from that hierarchy. These measures are discussed in more detail in Chapter 4 of this thesis.

Major and Mangano (1993) use a variety of interestingness criteria to evaluate and prune a set of discovered rules represented in the form *if A then B*. The first phase of the evaluation is the selection of *potentially interesting (PI)* rules from the initial set of discovered rules, by comparing the coverage and certainty factors of the rules. The *coverage G* of a rule is defined as the number of instances satisfying the concept A. The *certainty factor CF* of a rule is defined as the percentage of the covered instances in which a specified condition B is true. If rule 1 has $G = x_1$ and $CF = y_1$ and rule 2 has $G = x_2$ and $CF = y_2$, rule 1 is said to *dominate* rule 2 if $x_1 > x_2$ and $y_1 > y_2$. A rule that is not dominated by any other rule is said to be on the *performance frontier* and is considered to be a PI rule. The *cone* of a rule is defined as the set of specializations and generalizations of that rule. Rules in the cone of a PI rule are also considered to be PI rules. A rule which has no generalizations is called a *most general rule (MGR)* and the set of more specific rules in its cone is called the *family* of that MGR. If the family of a MGR contains three or more rules on the performance frontier, all rules in that family are considered to be PI.

The second phase of the evaluation is the selection of *technically interesting (TI)* rules from the set of PI rules. A technically interesting rule is one that is statistically unlikely to be a spurious specialization of another TI rule. The third stage of the evaluation is the removal of those rules in the set of TI rules that are redundant or similar but conflicting.

Agrawal, Imielinski and Swami (1993) examine transaction databases containing binary valued attributes to discover rules in the form $W \Rightarrow B$, where W is a set of one or more attributes and B is a single attribute not included in W . A minimum transaction support threshold, a minimum confidence factor threshold and syntactic constraints are used to limit the number of rules generated from a database. *Transaction support* is the percentage of the transactions that contain all attributes in W and attribute B . The *confidence factor* of a rule is defined as the percentage of the tuples in a relation for which the values of all attributes in W and the value of attribute B are 1. *Syntactic constraints* restrict or specify items appearing in the antecedent and the consequent of rules.

Klemettinen et al. (1994) extend the concept of syntactic constraints used by Agrawal, Imielinski and Swami (1993). *Templates* are used to specify interesting patterns to look for and uninteresting patterns to ignore in binary valued databases. A template can be used to restrict or specify antecedent and consequent attributes similar to syntactic constraints. However, the templates can also be used to restrict or specify classes appearing in the antecedent and the consequent of rules. A user-defined hierarchy maps database values into a taxonomy of classes. A commonness threshold is used to prune rules with low support and low confidence. The *commonness* of a rule is defined as the product of the confidence factor and the transaction support. Limits are also set on the number and size of rules.

This brief overview of some of the techniques for measuring the interestingness of discovery illustrates that the techniques are often domain specific, algorithm specific or

task specific. This is a reflection of the characteristics of the data upon which discovery is performed, the practical nature of knowledge discovery systems and the wide variety of applications to which knowledge discovery systems have been applied.

2.5 Inductive Learning and Knowledge Discovery Algorithms

Databases represent a store of useful, organized information. Information explicitly contained in the database can be efficiently retrieved using a database management system (DBMS). Information implicitly contained in a database must be obtained through the processes of deductive and inductive inference.

Deduction is a technique to infer information that is a logical consequence of the information in the database attributes (Holsheimer and Siebes, 1994). Given that the information in the database is correct, deductively inferred information represents provably correct facts about the real world. *Induction* is a technique to infer knowledge that is generalized from the information in the database attributes (Holsheimer and Siebes, 1994). Inductive inference provides information that is supported by the database but that is not necessarily true in the real world. Knowledge discovery systems attempt to obtain implicit information from the data in the database through the process of inductive inference.

In general, *inductive learning* can be defined as the process of acquiring knowledge by drawing inductive inferences from facts provided by a teacher or by the environment (Michalski, 1983). Machine learning systems attempt to automate inductive learning processes. Facts about the environment are provided through an input training

set. KDD algorithms also attempt to automate inductive learning processes, with facts about the environment available from a database.

The process of inductive learning can also be viewed as a search for plausible general descriptions (inductive assertions) that explain the given input data and are useful for predicting new data (Dietterich and Michalski, 1983). The *description space* is a set of descriptions partially ordered by relative generality, constructed from the input data and a description language. In a *top-down search* of the description space, an initial description chosen from the description space is specialized to find the best generalized description or descriptions according to some specified criteria. In a *bottom-up* or *data-driven search* of the description space, an initial description that covers the entire data set is repeatedly generalized to form a less complex but more general description.

Discovery of taxonomic descriptions, characteristic descriptions and discriminant descriptions are the most common tasks of knowledge discovery systems. *Taxonomic* or *classification* descriptions are descriptions that subdivide the set of input data into subclasses. The automatic detection of these subclasses is a problem of pattern identification or clustering. This is a form of *learning by observation and discovery* (Carbonell, Michalski and Mitchell, 1983), which is sometimes referred to as *unsupervised learning* (Frawley, Piatetsky-Shapiro and Matheus, 1991). In this learning method, no concepts are supplied. Observations of multiple aspects of the environment must be classified by the learner.

Once a pattern is detected, relevant qualities of the classes must be described. This process is a form of *learning from examples* (Carbonell, Michalski and Mitchell,

1983) and is sometimes referred to as *supervised learning* (Frawley, Piatetsky-Shapiro and Matheus, 1991). A set of examples and possibly counter examples of pre-classified concepts are provided. Using these examples, the learner must induce general concept descriptions that can be applied to correctly classify new examples. *Characteristic descriptions* state characteristics that are common to all known members of a class. *Discriminant descriptions* state facts that are necessary to distinguish a class from all other classes in a fixed set of classes.

Descriptions can also be classified as quantitative or qualitative, static or evolutionary. A *qualitative description* describes a logical relationship between attributes. A *quantitative description* describes a mathematical relationship between attributes. A *static description* describes the state of the data at the time of the discovery effort. An *evolutionary description* describes a change or trend in data.

2.6 Knowledge Representation

Knowledge representation is an important issue in the design of knowledge discovery systems, since it affects all aspects of the system. The representation of the input and output generalizations determines to a large extent the quality and utility of the output (Dietterich and Michalski, 1983). Some common representations include (Holsheimer and Siebes, 1994), (Dietterich and Michalski, 1983):

1. *Decision Tree*: a structure in which the nodes of the tree represent object attributes and edges leaving the nodes represent attribute values. The leaves of the tree

represent sets of objects with the same classification.

2. *Production rule*: a rules of the form $C \Rightarrow A$, where C is a set of conditions and A is an action taken if those conditions are met.
3. *Decision list*: a list of pairs of object descriptions and classes.
4. *Ripple-down rule set*: a nested set of if C then A rules.
5. *Semantic network*: a graph in which nodes represent concepts and the edges represent relationships between objects.
6. *Logic based expression*: logic expression whose components are propositions, predicates, variables or statements of variable restrictions.
7. *Frame*: a named structure containing information about an object.
8. *Taxonomy*: a hierarchy of concepts or objects.
9. *Neural network*: a densely interconnected network of elements, each of which contains a function to compute an output value from a set of inputs values.
10. *Relation*: a fixed set of attributes-domain pairs (columns) and a time varying set of tuples (rows) where each tuple is a set of attribute-domain value pairs.

The output from a discovery method is discovered knowledge represented in a form that is understandable and useful to the user. The representation of the discovered knowledge is dependent on who will use the discovered knowledge, the discovery algorithm used, the purpose of the discovered knowledge and the production of output that can be easily understood and utilized.

In the following chapter, a knowledge discovery method is described that represents input knowledge as a relation. Domain knowledge is incorporated into the method using a taxonomic representation. Finally, the knowledge discovered by the system is presented to the user in the form of a relation.

CHAPTER 3

ATTRIBUTE-ORIENTED GENERALATION (AOG)

3.1 Introduction

Generalization is the process of transforming a description so that it covers more objects. Two methods of generalization are the dropping condition operation and the climbing generalization tree operation (Michalski, 1983). The *dropping condition operation* removes a condition from a set of conjunctively linked conditions. For example, if the condition `car_type = "import"` is removed from the rule `IF car_type = "import" AND door_number = 2 AND driver_age < 18 THEN accident_cause = "Excessive Speed"`, the rule becomes more general since accidents involving non-import cars may now be covered by the rule. The *climbing generalization tree operation* replaces concepts with more general parent concepts. An example of this type of generalization is the replacement of the value "Saskatchewan" with a more general value "Prairies". A concept may be the parent of several less general concepts so that values "Manitoba" and "Alberta" may also generalize to the concept value "Prairies". *Attribute-oriented generalization (AOG)* (Cai, Cercone and Han, 1991) integrates database operations with these two generalization methods to generalize data from a database.

Section 3.2 provides a general description of attribute-oriented generalization using the terminology of Cai, Cercone and Han (1991). Section 3.3 gives an example of the application of AOG. Section 3.4 discusses the use of domain knowledge in the AOG process and Section 3.5 describes how the interestingness of results is considered. Section 3.6 summarizes strengths and weaknesses of the AOG process.

3.2 Description of Attribute-oriented Generalization

Attribute-oriented generalization transforms data stored in a database into more general information. A relation containing task specific data is retrieved from a database in response to a user defined discovery task. The task specific information is generalized according to a hierarchical classification of concepts, typically defined by a domain expert. The output presented to the user is a relation containing information generalized to a specified level.

Each attribute has an associated taxonomy of concepts called a concept hierarchy that guides the generalization of that attribute. A *concept hierarchy* is represented as a tree structure that has representations 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. Examples of concept hierarchies for two attributes are shown in Figure 3.1 and Figure 3.2. Concept hierarchies can be constructed for discrete attributes, as shown in Figure 3.1, or numeric attributes by mapping actual attribute values to a set of discrete values representing ranges of numeric values, as shown in Figure 3.2.

Figure 3.1: Concept Hierarchy for the Province Attribute

Figure 3.2: Concept Hierarchy for the Grant_Amount Attribute

The task relevant data values retrieved from the database are first converted to leaf concepts using the appropriate concept hierarchies. For example, using the concept hierarchy in Figure 3.2, a data value of 42021 for the Grant_Amount attribute would be converted to the leaf concept 30000-49999. The tuples containing the appropriate leaf concepts are inserted into a relation. This ungeneralized relation containing only task specific leaf concepts is referred to as the *target relation*. The target relation is then generalized by repeatedly replacing specific attribute values in the relation with more general concepts according to the defined concept hierarchies (generalization by climbing generalization trees). As related specific attribute values are grouped together into more general concepts, some tuples in the relation become redundant. By eliminating redundant tuples, the total number of tuples in the relation is reduced. If an attribute is generalized to the most general concept ANY, it is removed from the relation (generalization by dropping a condition).

The degree of generalization is governed by two user specified thresholds. An *attribute threshold* T_a for each attribute specifies the maximum number of distinct values for that attribute permitted in the final generalized relation. The *table threshold* T_t specifies the maximum number of tuples permitted in the final generalized relation. The generalization process first reduces the number of distinct values for each attribute to less than or equal to its attribute threshold. 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 attribute selection strategy and generalized to the next level in its concept

hierarchy. 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*. This is the relation that is presented to the user.

The number of tuples from the target relation that have been generalized to a given generalized tuple is called the *vote* of that generalized tuple. The votes are carried through the generalization process and are stored with the generalized tuples. For example, if a tuple carrying 5 votes and a tuple carrying 7 votes are generalized to identical tuples, then one of the identical tuples is eliminated and the other is assigned the 12 votes. The votes quantify the significance of the generalized tuples in the final generalized relation.

AOG is a data-driven generalization method that can be used to discover various types of descriptions from data in large databases. AOG was first developed to learn qualitative characteristic rules from relational databases (Cai, Cercone and Han, 1989). It has been extended to discover classification rules (Han, Cai and Cercone, 1991), discrimination rules (Han, Cai and Cercone, 1992), quantitative rules (Han, Cai and Cercone, 1993) and data evolution regularities (Han, Cercone and Huang, 1995). The AOG algorithm can be used to discover knowledge in nested relational or deductive databases (Han, Cai and Cercone, 1992), object-oriented databases (Nishio, Kawano and Han, 1993) and active databases (Han, Nishio and Kawano, 1993). A rule-based AOG method allows the incorporation of rules in concept hierarchy graphs to increase the representative and inductive power of the AOG process (Cheung, Fu and Han, 1994),

(Hwang and Fu, 1995). In this thesis, only the discovery of static characteristic rules in relational databases is considered. These rules provide both qualitative and quantitative information (Han, Cai and Cercone, 1993).

Due to the use of a DBMS to efficiently retrieve data from a database, efficient generalization algorithms (Carter and Hamilton, 1995a), (Carter and Hamilton, 1995b) and the ability to utilize domain knowledge in various forms, AOG is able to carry out the practical discovery of useful patterns in databases. Concept hierarchies allow the expression of patterns in terms of high level concepts and make AOG suitable for both discrete and numeric attribute values.

3.3 Example of Attribute-oriented Generalization

Table 3.1 is a tabular representation of a target relation containing leaf concepts for a Province attribute and a Grant_Amount attribute. The concept hierarchies for these attributes are shown in Figure 3.1 and Figure 3.2. Typically, not all leaf values in the hierarchies are active for a particular task. *Active* leaf concepts are those that are derived from the data retrieved from the database in response to a specified discovery task. Any concept that can be reached through ascension of the generalization tree from an active leaf concept is also defined as an active concept. In this case, all leaf concepts except Nfld. for the Province attribute and 10000-19999 for the Grant_Amount attribute are active. Assume $T_a = 4$ for the Province attribute, $T_a = 2$ for the Grant_Amount attribute and the table threshold $T_t = 6$.

Table 3.1: Sample Target Relation

Province	Grant Amount	Votes
B.C.	>100000	4
B.C.	90000-100000	8
B.C.	70000-89999	34
B.C.	50000-69999	41
B.C.	30000-49999	123
Ont.	>100000	9
Ont.	90000-100000	13
Ont.	70000-89999	53
Ont.	30000-49999	234
N.B.	30000-49999	38
N.S.	20000-29999	15
P.E.I.	30000-49999	6
P.E.I.	<10000	65
Alb.	90000-100000	4
Alb.	70000-89999	21
Man.	<10000	112
Sask.	<10000	98
Que.	>100000	12
Que.	90000-100000	43
Que.	70000-89999	65

B.C.	>100000	4
B.C.	90000-100000	8
B.C.	70000-89999	34
B.C.	50000-69999	41
B.C.	30000-49999	123

Generalize Concept Values

Western	>=90000	4
Western	>=90000	8
Western	50000-89999	34
Western	50000-89999	41
Western	20000-49999	123

Remove Duplicate Tuples

Western	>=90000	12
Western	50000-89999	75
Western	20000-49999	123

Figure 3.3: Illustration of the Generalization Process

To transform the target relation into the prime relation, the values in the relation are replaced with more general concepts through the ascension of the appropriate concept hierarchies and the elimination of duplicate tuples. Figure 3.3 illustrates this process for the first five tuples when "B.C." is replaced by "Western" and the resulting duplicate tuples are removed.

The relation represented in Table 3.2 is the prime relation. There are 4 distinct values of the Province attribute and 2 distinct values of the Grant_Amount attribute in the relation, so the attribute threshold criterion has been met for each attribute. Since there are 6 tuples in the relation and $T_i = 6$, the table threshold criterion has also been met. In this case, the prime relation is also the final generalized relation.

Table 3.2: Relation After Generalization to Attribute and Table Thresholds

Province	Grant_Amount	Votes	%Votes
Atlantic	Small grants	124	13
Ont.	Large grants	75	8
Ont.	Small grants	234	23
Western	Large grants	112	11
Western	Small grants	333	33
Que.	Large grants	120	12

3.4 Domain Knowledge

As previously discussed, domain knowledge in the form of concept hierarchies specified by a domain expert guides the generalization process. This makes a system using AOG flexible since the focus of the discovery effort can be easily changed by modifying the concept hierarchies or using different hierarchies. For example, in one concept hierarchy, concept values "Saskatchewan", "Manitoba" and "Alberta" might

generalize to a "Prairies" concept value, while in a different concept hierarchy, "Saskatchewan" and "Manitoba" might generalize to "low_population", while "Alberta" generalizes to "medium_population". The ability of the expert to supply and organize relevant high-level concepts plays a key role in the success of the discovery efforts.

Domain knowledge is also acquired through interaction with the user. The user specifies what data is to be retrieved from the database for a given discovery task. The user specifies which attributes are to be included in the generalization process and can restrict the data that is retrieved through queries to the DBMS. As a result, the usefulness of the discovered knowledge is affected by how well the user understands the domain that is described by that data. For example, if the user is trying to discover information relating patient symptoms to the diagnosis made, then selecting an attribute describing the amount of medical insurance that the patients carried is unlikely to assist in the discovery of interesting knowledge in the context of the current discovery effort. The user's domain knowledge assists in selecting appropriate data.

The user also controls the degree of generalization through the specification of the attribute and table thresholds. Unfortunately, domain knowledge is unlikely to provide adequate guidance about which threshold values will produce useful and interesting results. High threshold values may yield overly specific information which still hides interesting patterns. Low thresholds might result in overgeneralization and loss of interesting results. Finally, the user is required to interpret the output of the generalization process to determine if any interesting patterns exist. Again, the user's domain knowledge is required to identify patterns that are interesting.

3.5 Interestingness of Results

In a sense, the high level concepts in properly constructed concept hierarchies represent concepts that are considered interesting by a domain expert. The attributes and data chosen by the user for a discovery task represents what the user thinks might be interesting. However, the interestingness of the result is not directly considered during the generalization process.

Starting from the target relation, many possible generalized relations can be produced. Some of these relations may be more interesting to the user than others. A discovery system should be capable of deciding which calculations to perform and whether the results are interesting enough to constitute knowledge in the current context (Frawley, Piatetsky-Shapiro and Matheus, 1991). Currently AOG does not meet this requirement.

3.6 Strengths and Weaknesses of AOG

The primary strength of the attribute-oriented generalization process lies in the reduction of the complexity of the descriptions. Patterns between attributes in the relation may be easier to detect when the information is provided in a simpler form. Additionally, the more general concepts in the hierarchies represent a specification of what is interesting to the user. A shift in interest can be easily accommodated by changing the concept hierarchies. The problem of the size of the database is effectively handled by using a DBMS for efficient data retrieval and using efficient algorithms for the generalization process.

One weakness of the attribute-oriented generalization process is its reliance on user-specified thresholds, which require the user to choose thresholds that will produce useful and interesting information from the data. Additionally, interestingness is not explicitly considered during the generalization process. These weaknesses provide the motivation for the research described in this thesis. The goal was to examine the possible replacement of the user defined attribute threshold with a method for guiding the generalization process towards interesting results.

CHAPTER 4

ELIMINATION OF THE ATTRIBUTE THRESHOLD

4.1 Introduction

The attribute and table thresholds used in AOG are intended to specify and control the level of generalization (Han, Cai and Cercone, 1992). The domain knowledge of the user is unlikely to provide guidance concerning the specification of the appropriate thresholds that will produce interesting and useful results. Once the thresholds have been specified, the generalization proceeds with no direct consideration of the interestingness of the results that will be presented to the user. As a result, the discovery effort becomes a search for interesting results by trial and error specification of the thresholds. One way to solve this problem is to replace the attribute threshold with a method for guiding the generalization process towards interesting results.

Given an input relation, many different relations can be created by generalization. Let S be the set of relations including the input relation and all relations that can be produced by the generalization of one or more attributes of the input relation according to the concept hierarchies. Each relation in the set S can be represented in the terms of how many times each attribute has been generalized. For example, assume an input relation has three attributes a , b and c . A generalized relation $R_g(a_2b_3c_1)$ has been generated by generalizing attribute a two times, attribute b three times and attribute c once. Using this shorthand notation, the relationship between relations in the set S can be represented in

the form of an *attribute generalization graph*, where each node in the graph is a unique relation in S and each edge leaving a node represents a generalization of the node relation on a particular attribute.

Figure 4.1 shows a generalization graph for an input relation with three attributes. The node in the upper left hand corner of the graph represents the input relation. The node in the lower right hand corner of the graph is the maximally generalized relation. In the *maximally generalized relation*, all attributes have been generalized to the most general concept ANY. Moving from the input relation to the maximally generalized relation can be viewed as taking a path through the generalization graph, with each step in the path requiring the selection of an attribute to generalize. Applying AOG can be

Figure 4.1: Attribute Generalization Graph

regarded as following a path from the input relation towards the maximally generalized relation that stops as soon as an acceptable relation is reached.

If a single path is taken through the graph, some relations in the graph will not be generated. If each unique relation in the graph is assigned a value indicating its interestingness based on characteristics of the relation, it follows that the *attribute selection strategy*, the method used to select the next attribute for generalization, will affect the interestingness of the relations that can be generated.

In Section 4.2, criteria for evaluating an attribute selection strategy are discussed. In Section 4.3, attribute selection strategies that have been suggested in the literature are described and several new strategies are introduced. Examples are provided to illustrate the behavior of the different attribute selection strategies. In Section 4.4, a time complexity analysis of the strategies is presented.

4.2 Criteria for the Evaluation of Attribute Selection Strategies

4.2.1 Introduction

In general, the purpose of a KDD system 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 a level suited to human understanding. Two obvious criteria for evaluation of a strategy are its efficiency in selecting an attribute for generalization and its ability to produce interesting results. Another common criterion for KDD systems is the simplicity of the results that are presented to the user. A large and complex relation may contain interesting information, but if it cannot be understood by the user, it is not

useful and therefore uninteresting. One measure of simplicity is the number of tuples in a relation. For this reason, if two relations have the same value of interestingness measure, the one with fewer tuples is considered to be more useful and the behavior of the selection strategies is considered to be more important when there are few tuples than when there are many tuples.

4.2.2 Interestingness Criteria

While the efficiency of an attribute selection strategy can be easily evaluated, its ability to extract useful, interesting information and provide that information at an understandable level of detail is more difficult to measure and evaluate. As discussed in Section 2.4.2, various measures and methods have been employed in KDD systems to evaluate the interestingness of information discovered from databases. The two interestingness measures proposed by Hamilton and Fudger (1995) were selected for use in this study because they are directly based on the results of attribute-oriented generalization. Since the interestingness measures are based on the structure of the concept hierarchies that guide the generalization of attributes, no new domain knowledge is required. Additionally, the concepts and concept groupings in the concept hierarchies represent information that is considered interesting and important by the domain expert.

The interestingness measures assign an interestingness value to a generalized relation based on the interestingness of the attribute values in it. An attribute value in a relation is considered to be interesting only if it corresponds to an interior concept in the appropriate concept hierarchy. An *interior concept* is defined as any non-leaf, non-ANY

concept. The original values of the attributes and the ANY node of a concept tree do not provide any new information that cannot be obtained by direct database queries, so the occurrence of these values in a relation are considered to be uninteresting. For example, consider an attribute whose set of possible values are the Canadian provinces. A database query might yield the information that 10% of the tuples in a relation contain the attribute value “Sask.”, 20% contain the attribute value “Man.” and 25% contain the attribute value “Alta.”. This information is explicitly contained in the data and therefore it is not interesting discovery produced by the KDS. However, the information that 55% of the attribute values can be generalized to the interior concept “Prairies” is interesting because it is useful information that is not explicitly contained in the data.

Measure I_1 is the number of attribute values in an output relation that correspond to interior concepts in the appropriate concept hierarchies. Occurrences of the same attribute value are counted separately. The set of interior concept values in output relation R is represented by V_R and is defined by Hamilton and Fudger (1995) as

$$V_R = \{(t, a, R_{t,a}) \mid t \in R, a \in A, R_{t,a} \neq \text{ANY and } R_{t,a} \text{ is not a leaf}\} \quad (4.1)$$

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 defined as the cardinality of the set V_R :

$$I_1(R) = |V_R| \quad (4.2)$$

The second measure of interestingness, referred to as I_2 , also considers the depths and the weighted heights of concepts in the appropriate concept hierarchy. 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 descendants it has and a weight value. The weight value of a node is the sum of the distances to each of its leaf descendants.

The number of leaf descendants of a node in a concept hierarchy is determined using the equations

$$n(t) = 1 \quad \text{if } t \text{ is a leaf node} \quad (4.3)$$

$$n(t) = \sum_{c \in C(t)} n(c) \quad \text{otherwise} \quad (4.4)$$

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

$$w(t) = n(t) + \sum_{c \in C(t)} w(c) \quad (4.5)$$

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

$$wh_t = w(t)/n(t) \quad (4.6)$$

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

$$IN(v, cht) = (k)d_{cht}(v) + (1 - k)wh_{cht}(v) \quad (4.7)$$

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. 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. Choosing $k = 0.5$ weights these factors equally and is used henceforth.

Finally, the interestingness measure I_2 of the relation R is determined by the equation

$$I_2(R) = \sum_{(t, a, v) \in V_R} IN(v, tree(a)) \quad (4.8)$$

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

4.3 ATTRIBUTE SELECTION STRATEGIES

4.3.1 Introduction

A number of strategies for selecting the next attribute to generalize in AOG have been suggested in the literature. These include selection based on:

1. the greatest reduction in the number of tuples (Cai, Cercone and Han, 1991), (Han, Cai and Cercone, 1992), (Carter, Hamilton and Cercone, 1994)
2. the least reduction in the number of tuples (Carter, Hamilton and Cercone, 1994)
3. producing the simplest generalized relation (Cai, Cercone and Han, 1991), (Han, Cai and Cercone, 1992)
4. the number of distinct attribute values (Han, Cai and Cercone, 1992), (Carter, Hamilton and Cercone, 1994)
5. the ratio of distinct attribute values to T_a (Shan, Hamilton and Cercone, 1995)

No previous comparison of these methods has appeared in the literature, other than a initial report on the results of this research (Barber and Hamilton, 1996). In this section, the strategies suggested in the literature are discussed and several new strategies are introduced. Examples are provided to illustrate the behavior of the different attribute selection strategies.

A sample relation consisting of concept values for two attributes is shown in Table 4.1. Figure 4.2 depicts the concept hierarchies for the attributes in a tab format, where each tab represents one level deeper in the tree. In this example, all leaf concepts

Table 4.1: Sample Relation

Attribute A	Attribute B
Stats1	B.C.
Stats2	B.C.
Prob2	B.C.
Math1	B.C.
Math2	B.C.
Stats2	N.B.
Math2	N.B.
Stats2	N.S.
Stats2	P.E.I.
Prob2	P.E.I.
Stats1	Prairies
Prob1	Prairies
Math1	Prairies
Math2	Prairies

ANY

Allmath
 Stats
 Stats1
 Stats2
 Prob
 Prob1
 Prob2
 Math
 Math1
 Math2
 AllEnglish
 Lit
 Lit1
 Lit2
 Comp
 Comp1
 Comp2

ANY

Canada
 Ont.
 Que.
 Western
 B.C.
 Prairies
 Alta.
 Sask.
 Man.
 Atlantic
 N.B.
 N.S.
 Nfld.
 P.E.I.

a) Concept Hierarchy for Attribute A

b) Concept Hierarchy for Attribute B

Figure 4.2: Concept Hierarchies for Sample Relation Attributes

except for Lit1, Lit2, Comp1 and Comp2 are active. Table 4.2 and Table 4.3 show tabular representations of the relations that would be obtained by generalizing attribute A and attribute B, respectively. Assume $T_t = 5$ and $T_a = 6$ for both attributes.

Table 4.2: Relation R_A

Attribute A	Attribute B
Stats	B.C.
Prob	B.C.
Math	B.C.
Stats	N.B.
Math	N.B.
Stats	N.S.
Stats	P.E.I.
Prob	P.E.I.
Stats	Prairies
Prob	Prairies
Math	Prairies

Table 4.3: Relation R_B

Attribute A	Attribute B
Stats2	Atlantic
Prob2	Atlantic
Math2	Atlantic
Stats1	Western
Stats2	Western
Prob1	Western
Prob2	Western
Math1	Western
Math2	Western

4.3.2 Attribute Selection Strategy Classification

Each strategy can be classified as either lookahead (L) or predictive (P). In a *lookahead strategy*, the current relation, which has m attributes, is used to generate m new relations. Each of the new relations is created by generalizing a different attribute in the current relation once. For example, using the representation described in Section 4.1, a relation $R_g(a_1b_3c_2)$ would be used to create three new relations $R_g(a_2b_3c_2)$, $R_g(a_1b_4c_2)$ and $R_g(a_1b_3c_3)$. Properties of the new relations are used to select the best attribute to use for the actual generalization. Only one-step lookahead strategies, where a decision is made after looking ahead one generalization, are considered in this study. In a *predictive strategy*, values that exist prior to the generalization are used to make the selection.

Table 4.4 summarizes the strategy classification. In the table, each strategy is assigned a code name for convenience. In addition to the strategies suggested in the literature, four strategies based on the interestingness measures discussed in the previous section were also examined. Strategies S9 and S10 are lookahead strategies that select the attribute for generalization which produces the highest interestingness measure in the generalized relation. Strategy 9 uses I_1 as the interestingness measure and S10 uses I_2 . Strategies S4 and S5 are predictive strategies that use complexity measures developed by Hamilton and Fudger (1995) to select the attribute whose concept hierarchy has the greatest potential for interestingness. Strategies S4 and S5 are described in more detail in Section 4.3.4.2.

Table 4.4: Attribute Selection Strategy Classification

Attribute Selection Strategy	S-code	Class
Most Distinct Attribute Values	S1	P
Greatest Reduction in Tuples	S2	L
Least Reduction in Tuples	S3	L
Largest Complexity Measure M_1	S4	P
Largest Complexity Measure M_2	S5	P
Greatest R_{n/T_a}	S6	P
Least R_{n/T_a}	S7	P
Fewest distinct Attribute Values	S8	P
Largest Interestingness Measure I_1	S9	L
Largest Interestingness Measure I_2	S10	L

4.3.3 Lookahead Strategies

4.3.3.1 Selection Based on the Number of Tuples after Generalization (S2 and S3)

Adopting the strategy of selecting an attribute whose generalization will result in the greatest reduction in the number of tuples results in a relatively fast generalization to meet the table threshold criterion. However, if the number of tuples in a relation is just

slightly above the table threshold, then a large reduction in the number of tuples is not necessarily desirable. The generalization could generate a relation with a number of tuples that is farther away from the threshold.

Table 4.2 shows that a generalization of attribute A would reduce the number of tuples to 11. Table 4.3 shows that a generalization of attribute B would result in a relation with 9 tuples. Under this strategy, attribute B 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 the specified table threshold. However, the effort needed to attain that level of generalization would be relatively high.

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

4.3.3.2 Selection Based on Interestingness Measure I_1 and I_2 (S9 and S10)

Since one measure of the effectiveness of an attribute selection strategy is the interestingness measures I_1 and I_2 , a lookahead strategy that produces the generalization to the most interesting relation seems promising.

As an example of how the interestingness of each concept in a relation is determined, the concept Western for attribute B will be examined. The concept is an interior concept, so it is considered in the determination of the interestingness measures. There are six instances of the concept in relation R_B , so it adds 6 to the interestingness

measure I_1 . An examination of the concept hierarchy for attribute B shows that $d_{cht}(Western) = 2$ and that the concept has children B.C. and Prairies. The concept B.C. is a leaf concept so $n(B.C.) = 1$ and $w(B.C.) = 0$. The concept Prairies has 3 leaf concepts as children, each with $n = 1$ and $w = 0$. Therefore, $n(Prairies) = 1 + 1 + 1 = 3$ and $w(Prairies) = (1 + 1 + 1) + 0 = 3$. Now $n(Western) = 1 + 3 = 4$, $w(Western) = (1 + 3) + (0 + 3) = 7$ and $wh(Western) = 7/4 = 1.75$. Using $k = 0.5$, $IN(Western,cht) = 0.5(2) + 0.5(1.75) = 1.9$. Since there are six instance of Western in the relation, it contributes $6(1.9) = 11.4$ to the interestingness measure I_2 .

The values used to determine the interestingness measures for R_A and R_B are summarized in Table 4.5. In this table, i is the number of times a concept appears in a

Table 4.5: Values for Determination of Interestingness Measures

Values for Relation R_A					Values for Relation R_B				
Concept	Type	i	IN	i*IN	Concept	Type	i	IN	i*IN
Stats	Interior	5	1.5	7.5	Stats1	Leaf	-	0	-
Prob	Interior	3	1.5	4.5	Stats2	Leaf	-	0	-
Math	Interior	3	1.5	4.5	Prob1	Leaf	-	0	-
B.C.	Leaf	-	0	-	Prob2	Leaf	-	0	-
N.B.	Leaf	-	0	-	Math1	Leaf	-	0	-
N.S.	Leaf	-	0	-	Math2	Leaf	-	0	-
P.E.I.	Leaf	-	0	-	Atlantic	Interior	3	1.5	4.5
Prairies	Interior	3	1	3.0	Western	Interior	6	1.9	11.4
				$\Sigma = 14$					$\Sigma = 19.5$
									$\Sigma = 9$
									$\Sigma = 15.9$

relation. Using the table values, $I_1(R_A) = 14$, $I_2(R_A) = 19.5$, $I_1(R_B) = 9$ and $I_2(R_B) = 15.9$.

Since both I_1 and I_2 for R_A are greater than the corresponding interestingness measures

for the relation R_B , attribute A would be chosen for generalization regardless of which interestingness measure is used to make the selection.

4.3.4 Predictive Strategies

4.3.4.1 Selection Based on the Number of Distinct Attribute Values (S1 and S8)

Alternatives that select an attribute in a relation with the largest or smallest number of distinct attribute values (S1 or S8), or the largest or smallest ratio of distinct attribute values to attribute threshold (S6 or S7), have the advantage of simplicity. The calculations are simple and no lookahead is required. However, they do not consider the properties of the concept hierarchies or the actual generalization behavior.

Either S7 or S8 essentially chooses an attribute for repeated generalization until the table threshold is met or until it is maximally generalized. If the attribute becomes maximally generalized before the table threshold criterion is met, another attribute is chosen and the process repeated. As a result, any generalized relation has only one attribute with non-leaf attribute values. The remainder of the attributes in the relation have only leaf concept values.

Strategies S6 and S7 depend on the attribute threshold. As previously discussed, an appropriate attribute threshold that will produce interesting and useful results is difficult to estimate.

If the number of distinct values for attribute i is represented by N_i and the ratio of distinct attribute values to attribute threshold is represented by R_{N/T_a} , then in the sample relation, $N_a = 6$ and $N_b = 5$. Since $T_a = 4$ for both attributes, $R_{N/T_a} = 1.50$ for attribute A

and $R_{N/Ta} = 1.25$ for attribute B. If the largest N or $R_{N/Ta}$ were used as the selection criterion, then attribute A would be chosen for generalization. If the smallest N or $R_{N/Ta}$ were used, then attribute B would be chosen.

4.3.4.2 Strategies Based on the Complexity of the Hierarchy Trees (S4 and S5)

To estimate the potential for knowledge discovery from a database, heuristic measures have been developed that are closely related to the interestingness measures I_1 and I_2 (Hamilton and Fudger, 1995). The first measure of complexity, referred to as M_1 , is simply the sum of the interior concepts in a concept tree t . This is indicated by the equation

$$M_1 = \text{nin}_t \quad (4.9)$$

where nin_t is the number of interior concepts in the tree. The second measure of the complexity, referred to as M_2 , also considers the interestingness of the interior nodes of the tree. This measure is determined by the equation

$$M_2 = \sum IN(\text{value}(j,t),t) \quad (4.10)$$

where t is a concept tree, j is a node in the concept tree, $\text{value}(j,t)$ is the concept value of the node and $IN(\text{value}(j,t),t)$ is the interestingness of the node as defined in Section 4.2.2.

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 M_1 and M_2 to provide a guide for selecting an attribute for generalization. However, the properties of the actual attribute values in the relation being examined should also be considered. In strategies S4 and S5, only interior concepts in the concept tree with a depth less than that of the concepts in the current relation contribute to the complexity. In this way, only the concepts that may potentially be reached through further generalization are considered in the selection of the next attribute to generalize. Additionally, inactive branches are removed prior to the determination of the interestingness.

For attribute selection strategies based on M_1 (S4) and M_2 (S5), the interestingness of all potentially active concepts is determined once for a discovery task. For attribute B, assume that each of the concepts Alta., Sask., and Man. were active and generalized to the concept Prairies in the sample relation. Since the concept Nfld. is not found in the input relation, it is not active. Therefore, it is not considered in the calculation of the interestingness of the concepts Atlantic and Canada. For attribute A, the concepts Lit, Comp, English and AllEnglish are inactive.

The values used to determine the complexity measures for the sample relation attributes are summarized in Table 4.6. Using these values, $M_1(A) = 4$, $M_2(A) = 6.0$, $M_1(B) = 3$ and $M_2(B) = 5.1$. Since both $M_1(A)$ and $M_2(A)$ are greater than $M_1(B)$ and $M_2(B)$, respectively, attribute A would be chosen for generalization regardless of which complexity measure is used to make the selection.

Table 4.6: Values for Determination of Complexity Measures

Potentially Active for Attribute A			Potentially Active for Attribute B		
Concept	i	IN	Concept	i	IN
Stats	1	1.5	Atlantic	1	1.5
Prob	1	1.5	Western	1	1.9
Math	1	1.5	Canada	1	1.7
AllMath	1	1.5			
$\Sigma = 4$			$\Sigma = 3$		
$\Sigma = 6.0$			$\Sigma = 5.1$		

4.4 Time Complexity Analysis

The efficiency of an attribute selection strategy has two components. The first component is the number of generalizations that are required to produce a generalized relation that satisfies the table threshold criterion. The second component is the efficiency of the attribute selection for each of these generalizations.

Consider the target relation R_t with m attributes. Each attribute a has an associated concept hierarchy CH_a . Let $dmax_a$ be the maximum depth of the concept hierarchy for attribute a . By definition, the attribute values in the target relation are leaf concept values. Given an unbalanced concept hierarchy, the depth d_c of a leaf concept c in the target relation has limits $1 \leq d_c \leq dmax$. Now let c_a be a leaf concept value in the target relation for attribute a and d_{ca} be the depth of the concept in CH_a . If c_a is the concept value for attribute a with the greatest depth in CH_a , then attribute a can be generalized at most d_{ca} times. The maximum number of possible generalizations of the target relation g_{max} is then

$$g_{max} = \sum_{a=0}^{a=m-1} d_{ca} \quad (4.11)$$

The actual number of generalizations g_{actual} depends on the concept values in the target relation, the table threshold and the structure of the involved concept hierarchies.

In the lookahead strategies, the input relation must be generalized on each of its m attributes. Each of the n_i tuples in the input relation must be examined, the appropriate concept generalized one level to create a new tuple and any unique new tuple inserted into a new relation. The cost to carry out the actual generalization by tree ascension is relatively small for large relations compared to the cost to create the new relation (Carter 1994). To determine if a tuple is unique, it is compared to the tuples previously inserted in the new relation. If the new relation has n_o tuples, the comparison and insertion of each tuple would take at most mn_o steps. The creation of a new relation would take at most mn_in_o steps. Therefore, the generalization of a single attribute to create a new relation is $O(mn_in_o)$. The time complexity to carry out this generalization and relation creation for m attributes is $O(m^2n_in_o)$ 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 (S2 and S3), no further work is required except for the comparison of the number of tuples in the new relations. This is done as the new relations are created, so the time complexity of these strategies is $O(m^2n_in_o)$.

For the lookahead strategy based in the interestingness measures I_1 and I_2 , the interestingness of m generalized relations must be determined. This involves examining each concept in a relation and calculating the interestingness of any interior concepts. The process of identifying interior concepts can be done using existing information

associated with the concepts in the relations, so mn_o steps are required for each relation or m^2n_o steps for m new generalized relations. For I_1 , a count of the interior concepts for each new relation is sufficient to make a selection, so the complexity of strategy S4 is $O(m^2n_in_o + m^2n_o)$ or $O(m^2n_in_o)$. For the strategy based on I_2 , the interestingness value of each concept in the concept hierarchies must be calculated. However, the interestingness values depend only on the structure of the concept hierarchies, so these calculations need only be done once for all tasks involving those hierarchies. As a result, the time complexity of strategy S5 is also $O(m^2n_in_o)$.

In the predictive strategies, some properties of the input relation is used to determine which attribute to generalize. For the strategies based on the number of distinct attribute values or R_{NTa} 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 $O(m)$.

For the strategies based on the complexity of the remaining hierarchy trees, inactive concepts must be pruned from the concept hierarchies. This operation must be done once for each discovery task, since the values in the target relation determine which concepts will be active. A list of distinct attribute values for each attribute is stored as the target relation is created. For each concept in a distinct concept list, the path to the most general concept ANY is traveled. Any distinct concept values encountered are inserted into a list of potentially active concepts. If the maximum depth of a concept in a hierarchy is $dmax$, then at most $dmax - 1$ concepts are examined on a path to the ANY concept. If there are m_d concepts in the initial list, then at most $(m_d(dmax-1))$ concepts

are inserted into the list. The time complexity to create a list of distinct potentially active concepts is $O(m_d^2 dmax)$.

After the list of potentially active concepts is created, the concept hierarchy must be traversed to determine the values needed for the calculation of the complexity measures. If a concept hierarchy contains h concepts, then the concept tree traversal is $O(h)$. During the traversal, summaries of values for active concepts above each level in the hierarchy can be compiled. Therefore, the time complexity to create the summaries for an attribute is $O(h)$. The time complexity to create the list of potentially active concepts and to compile the summaries for an attribute is $O(m_d^2 dmax+h)$ and $O(m(m_d^2 dmax+h))$ for all attributes in the target relation. Using the pre-compiled summaries, the time complexity to select the attribute with the most complex remaining hierarchy is $O(m)$.

A summary of the complexity analysis is shown in Table 4.7.

Table 4.7: Summary of Time Complexity Analysis for a Single Generalization

Strategies	Class	Time Complexity
S1, S4, S5, S6, S7, S8	P	$O(m)$
S2, S3, S9, S10	L	$O(m^2 n_i n_o)$

CHAPTER 5

ANALYSIS AND DISCUSSION OF EXPERIMENTAL RESULTS

5.1 Introduction

In this chapter, the experimental methodology is described and the experimental results are presented, evaluated and discussed. Section 5.2 describes the DB-Discover knowledge discovery program, which was used to test the attribute selection strategies, and the experimental platform. Section 5.3 provides details about the databases upon which the strategies were tested. Section 5.4 outlines the experimental procedure and test program. In Section 5.5, the method used to evaluate the strategies is described. Section 5.6 discusses the general behavior of the lookahead strategies S2, S3, S9 and S10. In Section 5.7, the predictive strategies S1, S4, S5 and S8 are evaluated and discussed. Section 5.8 provides an overall comparison of the strategies. Finally, Section 5.9 describes a modification of strategies S9 and S10 that was tested in response to the experimental results obtained with the original strategies. The modified strategies are compared to the best of the original strategies.

5.2 The DB-Discover Program

DB-Discover is a knowledge discovery program that uses attribute-oriented generalization to discover new and nontrivial information that is implicitly contained in

the data of a database (Rivera and Carter, 1995). The DB-Discover program generalizes all attributes to less than or equal to their respective attribute thresholds and then, if necessary, uses the predictive strategy of selecting the attribute with the most distinct values (S1) for further generalization from the prime relation to the final generalized relation. The user also has the option of selecting an attribute for further generalization.

To evaluate and compare the behavior of the attribute selection strategies discussed in the previous chapter, strategies S2, S3, S4, S5, S8, S9 and S10 were implemented and linked to DB-Discover. Strategies S6 and S7 use the attribute threshold and, since the elimination of the attribute threshold is one goal of this thesis, these strategies were not tested.

The user interface of DB-Discover was bypassed to facilitate and automate the testing process. Multiple task specifications were obtained from a file and all results were written to file. A program was written to generate multiple tasks by specifying a database name, a query predicate, the join tables and a list of attributes. All tasks in a generated task specification file shared a common database name, query predicate and join tables. However, each individual task specification contained a unique combination of two or more attributes so that the target relation retrieved for a task would be unique.

The experimental data was obtained using a client-server version of DB-Discover suited to a distributed computing environment. The client machine was a Silicon Graphics Challenge M workstation with 256 MB of memory and eight 150 MHz MIPS R440 CPUs running an IRIX Release 5.3 UNIX operating system. The server portion of DB-Discover was installed on a SUN IPX workstation, running a SUN Release 4.1.3

operating system. The databases were accessed through an Oracle Version 7.1.1.16 DBMS running on the SUN IPX workstation.

The large size of one of the test databases required modification to the server portion of the DB-Discover program to cache retrieved data and eliminate multiple database queries for tasks with identical query predicates and join tables. This modification reduced the time required to run a discovery task by approximately 75%.

5.3 Description of Test Databases

Testing was done using two databases. The first was a public domain National Science and Engineering Research Council of Canada (NSERC) grant and award information database. It is a relatively small 10,000 tuple database with 6 tables describing 22 attributes. Simple concept hierarchies were available for four of its attributes (Carter, Hamilton and Cercone, 1994).

The second database was a commercial 8,000,000 tuple database provided by Saskatchewan Telecommunications (SaskTel). The database consists of 22 tables describing 56 attributes, with table sizes ranging from 22 tuples to 3,300,000 tuples. Complex and detailed hierarchies were developed for 31 of the attributes using information provided by SaskTel. While the tab format hierarchy files for the NSERC database contained 440 lines, the hierarchy file for the SaskTel database contained 7400 lines.

5.4 Test Program and Experimental Procedure

In total, the strategies were tested on 638 different discovery tasks on the two test databases. Since the SaskTel concept hierarchies were larger and much more complex than the NSERC concept hierarchies, they provided far greater opportunity for diversity in the specification of discovery tasks. For this reason, 475 of the discovery tasks were performed on the SaskTel database, while 163 discovery tasks were performed on the NSERC data. The experimental procedure described in the remainder of this section was followed for each of the target relations retrieved for these tasks.

After the target relation was retrieved for a task, all unique relations that could be produced through one or more generalizations of the attributes in the target relation were generated. I_1 , I_2 , n and the number of times each attribute was generalized were stored for each of these relations. Additionally, I_1max_n and I_2max_n were determined. I_1max_n and I_2max_n are the maximum I_1 and I_2 values of relations with n tuples in the set of all possible unique relations for a task. Each target relation was then generalized step by step to a maximally generalized relation using each of the selection strategies. After each generalization step, information about the newly generalized relation was stored.

Table 5.1 provides an example of the information that was stored for a complete generalization from a target relation to a maximally generalized relation for each attribute selection strategy. The first row in the table represents the target relation retrieved for a specific task. In this case, the target relation has $n = 114$ and $m_a = 4$, where m_a is the number of active attributes. An *active attribute* is defined as one that is not generalized to the ANY concept. Each subsequent row i in the table contains information about one generalized relation Rg_i . The second row represents the initial generalization of the target

relation, the third row is the generalization of the relation described in the second row and so on. The last row represents the maximally generalized relation, which always consists of a single tuple and no active attributes. The value $index_i$ is the index of the attribute that

Table 5.1: Sample Test Results for One Attribute Selection Strategy

index	n	m _a	I ₁	I ₁ max _n	%d ₁	I ₂	I ₂ max _n	%d ₂
N/A	114	4	0	N/A	N/A	0	N/A	N/A
1	114	4	81	309	-74	120	459	-74
1	99	4	99	297	-67	130	424	-69
0	99	4	198	297	-33	277	424	-35
0	71	4	142	222	-36	195	318	-39
1	45	3	45	90	-50	66	132	-50
3	41	3	48	89	-46	74	134	-45
3	31	3	43	104	-59	64	150	-57
0	11	2	5	16	-69	6	22	-73
3	3	2	3	6	-50	3	6	-50
2	3	2	6	6	0	6	6	0
2	1	1	1	2	-50	0	0	0
3	1	0	0	N/A	N/A	0	N/A	N/A

was generalized to create Rg_i , %d₁ is the percent difference between the computed I_1 of Rg_i and I_1max_n and %d₂ is the percent difference between the computed I_2 of Rg_i and I_2max_n . The percent difference values are calculated using the equations $\%d_1 = (I_1 - I_1max_n)/I_1max_n$ and $\%d_2 = (I_2 - I_2max_n)/I_2max_n$. By definition, $I_1 = 0$ and $I_2 = 0$ for both the target relation and the maximally generalized relation, so I_1max_n , I_2max_n , %d₁ and %d₂ are not applicable to these relations. The target relation is ungeneralized so $index_i$ is not applicable to the target relation. Real values of I_2 were stored as integers, so the decimal digits of the computed I_2 values were truncated.

5.5 Method of Evaluation

One method of comparing the behavior of the strategies during the generalization

process is to use charts such as those shown in Figure 5.1. Figure 5.1 plots the calculated interestingness measures I_1 and I_2 against the number of tuples in the relations produced by each strategy for the SaskTel task T27. This task is representative of the behavior of the strategies for all tasks and it will be used to illustrate strategy behavior throughout this chapter. While charts such as those in Figure 5.1 can be used to detect general behavior patterns of the strategies for individual tasks, there are several problems with this approach. First, overlapping data for the different strategies can obscure behavior patterns. Secondly, the different strategies produce relations of differing sizes, so a direct comparison of the strategies for a particular number of tuples may not be possible. Finally, this method of comparison does not provide a quantitative measure of the effectiveness of the strategies.

Also shown in Figure 5.1 are I_2max_n and I_1max_n values for the discovery task. As described in Section 5.4, these values represent the maximum I_1 and I_2 values of all possible relations with n tuples for a task. These values exhibit scatter because the relations corresponding to the I_1max_n or I_2max_n values for a task frequently lie on different paths through the attribute generalization graph. As a result, none of the strategies examined here can produce all the maxima for a task. However, these values provide an upper bound that can be used as an indication of the effectiveness of the various strategies in producing interesting results. The performance of a strategy for a particular task is quantified by calculating $\%d_1$ and $\%d_2$, the percent difference between the actual interestingness measures of a relation with n tuples produced by the strategy and the corresponding I_{max_n} value.

Figure 5.1: I_1 and I_2 versus Number of Tuples for all Strategies in Task T27

$\%d_1$ and $\%d_2$ were determined for thousands of relations with sizes ranging from 3500 tuples to 1 tuple. Since the behavior of the strategies is considered to be more important for smaller, simpler relations that are more likely to be understandable to the user, the discussion in this chapter focuses on relations with 100 or fewer tuples. As data was collected for approximately 52,000 relations with a size within this range, it was necessary to summarize and simplify the data to facilitate analysis. This was done by calculating $\%d_{1a}$ and $\%d_{2a}$, the average $\%d_1$ and $\%d_2$ over ranges of tuples for all relations produced by each strategy in all tasks. Figure 5.2 provides an example of a chart of $\%d_{1a}$ versus tuples range that is used to examine the general performance of the individual strategies. The horizontal axis of the chart is labeled with ten equal tuple ranges from 1-10 to 91-100. The $\%d_{1a}$ or $\%d_{2a}$ value for a range represents the average $\%d_1$ or $\%d_2$ values of hundreds or thousands of relations with a size in the range.

Figure 5.2: Example chart of $\%d_{1a}$ versus Tuple Range

As shown in Figure 5.2, separate $\%d_{1a}$ and $\%d_{2a}$ values were calculated for the SaskTel and NSERC tasks to permit comparison of the general behavior of the strategies in discovery tasks performed on the two different databases.

Although both I_1 and I_2 were calculated for all relations, the indicated behavior of the strategies was found to be similar regardless of whether I_1 or I_2 was used to determine the interestingness of the relations. For the most part, I_1 will be used to illustrate strategy behavior in this chapter. Similarly, the behavior indicated by $\%d_{1a}$ and $\%d_{2a}$ are similar, so $\%d_{1a}$ is used to illustrate strategy behavior. Charts showing $\%d_{2a}$ versus tuple range for each of the strategies are included in Appendix A.

5.6 Analysis of the Lookahead Strategies

5.6.1 Analysis of Strategy S2

Figure 5.3 shows a plot of the I_1 values of relations produced in task T27 when the attribute selection strategy was based on producing the greatest reduction in the number of tuples. Table 5.2 provides information about the generalization process for this task.

Figure 5.3 shows that the I_1 values of the relations produced by strategy S2 for the sample task are well below the I_{1max_n} values for most of the generalization process, averaging -66% below maximum values over the entire task. Figure 5.4 shows the $\%d_{1a}$ values versus tuple range for all relations with 100 or fewer tuples produced by this strategy in all tasks. In relations with 11 to 100 tuples, the $\%d_{1a}$ values are approximately -50% for both the NSERC and SaskTel tasks. For relations with 1-10 tuples, there is a

Figure 5.3: Strategy S2 I_1 versus Number of Tuples for Task T27

Table 5.2: Strategy S2 Generalization Information for Task T27

index	n	m_a	I_1	I_{1max_n}	$\%d_1$
N/A	59	4	0	N/A	N/A
2	57	4	25	196	-87
2	55	4	26	191	-86
2	52	4	52	208	-75
2	33	4	33	122	-73
2	23	4	23	92	-75
2	19	4	19	69	-72
2	14	4	14	56	-75
0	14	4	28	56	-50
0	11	3	11	33	-67
1	11	3	16	33	-52
1	9	3	14	23	-39
1	7	3	14	21	-33
1	3	3	6	9	-33
1	1	3	2	3	-33
1	1	2	1	3	-67
2	1	1	0	3	-100
3	1	1	1	3	-67
3	1	1	1	3	-67
3	1	0	0	N/A	N/A

Figure 5.4: $\%d_{1a}$ versus Tuple Range for Strategy S2

significant decrease in the negativity of $\%d_{1a}$ values to approximately -37%. However, as will be discussed in Section 5.8, several other strategies perform better in this tuple range.

The poor performance of strategy S2 can be attributed to a tendency to generalize the same attribute many times in a row. For example, Table 5.2 shows that the first seven generalizations for the sample task are done on the same attribute. Since none of the other attributes have been generalized, they do not contribute to the interestingness of the relations, resulting in poor performance. The tendency to repeatedly generalize the same attribute is due to the relationship between the reduction in the number of tuples produced by the generalization of an attribute and the number of distinct values of that attribute after the generalization. Each distinct value of an attribute is a constraint to the elimination of redundant tuples. As the values of an attribute are generalized, the number of distinct values of the attribute in the relation is reduced. Each time an attribute is

generalized, the number of distinct values is further reduced, reducing the number of constraints and increasing the chance that the same attribute will again produce the greatest tuple reduction in the next generalization.

The repeated generalization of the same attribute increases the likelihood that it will be generalized to the ANY concept early in the generalization process. A relation with n tuples and m active has more potentially interesting concepts than one with n tuples and $m-1$ active attributes. As a result, strategies that delay the generalization of attribute values to the ANY concept until late in the generalization process are more likely to produce more interesting relations in the region of interest than those, such as S2, that eliminate attributes early in the process.

5.6.2 Analysis of Strategy S3

Figure 5.5 shows a plot of the I_I values of relations produced in task T27 when the attribute selection strategy was based on producing the least reduction in the number of tuples. Table 5.3 provides information about the generalization process for the task.

An examination of Figure 5.5 shows that strategy S3 was not effective in generating interestingness relations for the sample task. Using the values for $\%d_I$ in Table 5.3, the average $\%d_I$ over the entire task was -57%. For this task, strategy S3 displays the effect of early generalization of attributes to the ANY concept. Starting from an input relation with 59 tuples, the first attribute is maximally generalized while the relation still has 59 tuples, the second for a relation with 57 tuples and the third for a

Figure 5.5: Strategy S3 I_1 versus Number of Tuples for Task T27

Table 5.3: Strategy S3 Generalization Information for Task T27

index	n	m_a	I_1	I_{1max_n}	$\%d_1$
N/A	59	4	0	N/A	N/A
0	59	4	59	177	-67
1	59	4	89	177	-50
1	59	4	101	177	-43
1	59	4	118	177	-33
3	59	4	177	177	0
3	59	4	177	177	0
3	59	3	118	177	-33
0	57	2	57	196	-71
2	55	2	80	191	-58
1	52	2	76	208	-63
1	48	2	71	119	-40
1	48	1	23	119	-81
2	36	1	14	144	-90
2	29	1	29	87	-67
2	8	1	8	32	-75
2	4	1	4	16	-75
2	2	1	2	8	-75
2	1	1	1	3	-67
2	1	0	0	N/A	N/A

relation with 48 tuples. Since any subsequent relation can have only one attribute with values contributing to the interestingness of the relations, the performance is poor for the remainder of the generalization.

In the sample task, I_I initially increases with no change in the number of tuples because of one-to-one generalizations. In a *one-to-one generalization*, the concepts for an attribute are simply renamed at a higher level with no summarization taking place. Generally, one-to-one generalizations should be eliminated from concept hierarchies, since the generalization to a higher level adds no new summarization information (Hamilton and Fudger, 1995). However, in practice there may be one-to-one generalizations that add information. For example, if a part number 456YD generalizes to the English description “Brand X Black High Heels”, it can be argued that information has been added by the generalization. Data dependent one-to-one generalizations can also occur. For example, if an input relation has a PROVINCE attribute and all tuples retrieved from the database have a value of Sask. for the attribute, then a generalization to the next level will result in all values for the attribute becoming equal to the parent concept, with no new summarization information added.

Figure 5.6 shows the $\%d_{I_a}$ versus tuple range for all relations with 100 or fewer tuples produced by this strategy in all tasks. There is a clear trend for the $\%d_{I_a}$ values to become increasingly negative toward the end of the generalization process. This can again be attributed to the early generalization of concepts to the ANY concept.

Figure 5.6: $\%d_{1a}$ versus Number of Tuples for Strategy S3

5.6.3 Analysis of Strategies S9 and S10

Figure 5.7 shows a plot of I_I values of relations produced in task T27 for the attribute selection strategies based on the greatest interestingness measure I_I and greatest

Figure 5.7: Strategies S9 and S10 I_I versus Number of Tuples for Task T27

interestingness measure I_2 . Table 5.4 provides the task generalization information for S9 and Table 5.5 for S10.

As shown in Figure 5.7, the initial performance of these strategies is excellent. The interestingness of the relations produced by these strategies increases rapidly during the first several generalizations. As the initial generalization steps occur, leaf concepts are 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 tuple elimination. This is most noticeable for these strategies because the number of tuples in the relations reduces slowly and they experience the greatest increase in interestingness. Eventually, most leaf concepts are generalized and the loss of interestingness exceeds the gain in interestingness, resulting in a net interestingness decrease.

The $\%d_I$ values in Table 5.4 and Table 5.5 are somewhat misleading for the initial generalizations. For example, the input relation for the sample task has 59 tuples and $I_I = 0$, since by definition it contains only leaf concepts. After the first generalization using strategy S9, the relation still has 59 tuples and $I_I = 59$ since all of the leaf concepts in one column have been generalized to interior concepts. This is the maximum possible I_I value after one generalization. However, the $I_{I\max_n}$ value shows that some relation in the attribute generalization graph has 59 tuples and $I_I = 177$ and the resulting $\%d_I$ is -67%. The $\%d_I$ values for S9 and S10 drop sharply to 0% after the first several generalizations.

Table 5.4: Strategy S9 Generalization Information for Task T27

index	n	m _a	l ₁	l ₁ max _n	%d ₁
N/A	59	4	0	N/A	N/A
0	59	4	59	177	-67
3	59	4	118	177	-33
1	59	4	148	177	-16
2	57	4	168	196	-14
1	57	4	180	196	-8
1	57	4	196	196	0
3	57	4	196	196	0
2	55	4	191	191	0
2	52	4	208	208	0
1	46	4	184	184	0
1	36	4	144	144	0
1	36	3	108	144	-25
3	36	2	72	144	-50
0	29	1	29	87	-67
2	8	1	8	32	-75
2	4	1	4	16	-75
2	2	1	2	8	-75
2	1	1	1	3	-67
2	1	0	0	N/A	N/A

Table 5.5: Strategy S10 Generalization Information for Task T27

index	n	m _a	l ₁	%d ₁	%d ₁
N/A	59	4	0	N/A	N/A
2	57	4	25	196	-87
1	57	4	54	196	-72
3	57	4	111	196	-43
0	57	4	168	196	-14
1	57	4	180	196	-8
1	57	4	196	196	0
3	57	4	196	196	0
2	55	4	191	191	0
2	52	4	208	208	0
1	46	4	184	184	0
3	46	3	138	184	-25
1	36	3	108	144	-25
0	29	2	58	87	-33
1	29	1	29	87	-67
2	8	1	8	32	-75
2	4	1	4	16	-75
2	2	1	2	8	-75
2	1	1	1	3	-67
2	1	0	0	N/A	N/A

The performance of these strategies near the end of the generalization process is surprisingly poor. This results in part because these strategies eliminate attributes relatively early in the generalization process. In the sample task, strategy S9 maximally generalizes the first attribute when there is 36 tuples in the relation and strategy S10 maximally generalizes the first attribute at 46 tuples. The value of $\%d_1$ increases sharply after the attribute is eliminated. An examination of the data reveals that relations produced by other strategies that still have 4 active attributes are producing the I_1max_n and I_2max_n values.

An additional factor that may reduce the average effectiveness of these strategies for smaller relations is that if the initial target relation is large, the most interesting generalizations may occur before the size of the relations reduce to the point where they will be understandable to the user. However, if the initial target relation is small, then generalizing to produce the most interesting relations is more appropriate. Figure 5.8 and Figure 5.9 illustrate the trend for increasing negativity in $\%d_{1a}$ values for relations with fewer than 100 tuples, similar to the behavior of strategy S3.

Figure 5.8 and Figure 5.9 also show that the behavior of strategies selecting attributes based on the greatest I_1 (S9) and greatest I_2 (S10) are very similar. This is because most of hierarchies are *balanced trees*, where the none of the children of a node differ in depth below the parent node by more than 1, and the k term in equation 4.7 used to compute the IN value for an individual concept is set equal to 0.5. In such circumstances, all interior nodes have an equal IN value and a comparison of relations using I_1 is equivalent to a comparison using I_2 , although the magnitude of the values will be different. In fact, the calculated $\%d_{1a}$ and $\%d_{2a}$ values produced by these two

strategies are often not significantly different. The statistical analysis of the $\%d_{1a}$ and $\%d_{2a}$ values is discussed further in Section 5.8 and in Appendix B.

Figure 5.8: $\%d_{1a}$ versus Tuple Range for Strategy S9

Figure 5.9: $\%d_{1a}$ versus Tuple Range for Strategy S10

5.7 Analysis of the Predictive Strategies

5.7.1 Analysis of Strategy S1

Figure 5.10 shows a plot of I_I values of relations produced in task T27 when the attribute selection strategy was based on the selection of the attribute with the most distinct values. Table 5.6 provides the generalization information for the task.

Figure 5.10: Strategy S1 I_I versus Number of Tuples For Task T27

For the sample task, the interestingness measures are well below the maximum for most of the generalization process, averaging -63% over the entire task.

Figure 5.11 shows $\%d_{I_a}$ versus tuple range for all tasks. When the behavior of strategy S1 is examined in terms of all tasks, a slight improvement in the effectiveness of the strategy is displayed for relations with 20 or fewer tuples.

Table 5.6: Strategy S1 Generalization Information for Task T27

index	n	m _a	l ₁	l ₁ max _n	%d ₁
N/A	59	4	0	N/A	N/A
2	57	4	25	196	-87
2	55	4	26	191	-86
2	52	4	52	208	-75
2	33	4	33	122	-73
1	33	4	48	122	-61
1	33	4	56	122	-54
1	30	4	60	120	-50
2	20	4	40	80	-50
1	15	4	30	60	-50
2	10	4	20	40	-50
1	4	4	8	16	-50
0	4	4	12	16	-25
0	2	3	4	8	-50
2	1	3	2	3	-33
1	1	2	1	3	-67
2	1	1	0	3	-100
3	1	1	1	3	-67
3	1	1	1	3	-67
3	1	0	0	N/A	N/A

Figure 5.11: %d_{1a} versus Tuple Range for Strategy S1

5.7.2 Analysis of Strategy S8

Figure 5.12 shows I_I values of the relations produced in task T27 when the attribute selection strategy was based on selecting the attribute for generalization which has the fewest distinct values. Table 5.7 provides the generalization information for the task.

Strategy S8 performs poorly in the sample task, with an average $\%d_I$ over the entire task of -79%. This strategy acts to generalize and eliminate attributes one at time. The attribute with the fewest distinct values will have even fewer distinct values after the generalization, so it will be selected for the next generalization until it has been

Figure 5.12: Strategy S8 I_I versus Number of Tuples For Task T27

maximally generalized. Only the attribute being generalized can have interesting concepts, while values for all other attributes in the relation are uninteresting leaf or ANY concepts. As a result, the performance of this strategy is poor. The interestingness rises

Table 5.7: Strategy S8 Generalization Information for Task T27

index	n	m _a	l _i	l _i max _n	%d _i
N/A	59	4	0	N/A	N/A
3	59	4	59	177	-67
3	59	4	59	177	-67
3	59	3	0	177	-100
0	59	3	59	177	-67
0	57	2	0	196	-100
1	57	2	29	196	-85
1	57	2	41	196	-79
1	57	2	57	196	-71
1	54	2	54	186	-71
1	50	2	50	173	-71
1	50	1	0	173	-100
2	48	1	23	119	-81
2	36	1	14	144	-90
2	29	1	29	87	-67
2	8	1	8	32	-75
2	4	1	4	16	-75
2	2	1	2	8	-75
2	1	1	1	3	-67
2	1	0	0	N/A	N/A

as the attribute is generalized, and then falls to a zero value when the attribute is generalized to the ANY concept. In addition to producing uninteresting relations, strategy S8 presents the user with a mixture of generalized and ungeneralized information, which may be difficult to interpret.

Figure 5.13 shows the %d_{la} values versus the tuple range for relations with fewer than 100 tuples produced by strategy S8. The performance of this strategy is consistently poor for both the SaskTel tasks and the NSERC tasks. The %d_{la} values range from -66% to -75% for SaskTel tasks and -69% to -80% for NSERC tasks.

Figure 5.13: $\%d_{1a}$ versus Tuples Range for Strategy S8

5.7.3 Analysis of Strategies S4 and S5

Figure 5.14 shows a plot of I_l values of the relations produced in task T27 when the attribute selection strategies were based on the selection of the attribute with the greatest complexity measure M_1 and greatest complexity measure M_2 . Table 5.8 gives the task generalization information for strategy S4 and Table 5.9 provides similar information for strategy S5.

In the sample task, both strategy S4 and S5 perform poorly near the start of the generalization process, with an average $\%d_1$ value for relations with greater than 10 tuples of -59% for strategy S4 and -57% for strategy S5. However, both strategies perform better near the end of the generalization process. For relations with fewer than 10 tuples the relations produced by strategy S4 and strategy S5 have an average $\%d_1$ of -12% and -31% respectively.

Figure 5.14: Strategies S4 and S5 I_1 versus Number of Tuples for Task T27

Table 5.8: Strategy S4 Generalization Information for Task T27

index	n	m_a	I_1	I_{1max_n}	% d_1
N/A	59	4	0	N/A	N/A
2	57	4	25	196	-87
2	55	4	26	191	-86
2	52	4	52	208	-75
1	52	4	78	208	-62
1	52	4	90	208	-57
2	33	4	56	122	-54
1	30	4	60	120	-50
2	20	4	40	80	-50
1	15	4	30	60	-50
2	10	4	20	40	-50
0	10	4	30	40	-25
3	10	4	40	40	0
1	4	4	16	16	0
2	2	4	8	8	0
3	2	4	8	8	0
0	1	3	3	3	0
1	1	2	2	3	-33
2	1	1	1	3	-67
3	1	0	0	N/A	N/A

Table 5.9: Strategy S5 Generalization Information for Task T27

index	n	m _a	l ₁	l ₁ max _n	%d ₁
N/A	59	4	0	N/A	N/A
2	57	4	25	196	-87
2	55	4	26	191	-86
2	52	4	52	208	-75
2	33	4	33	122	-73
2	23	4	23	92	-75
2	19	4	19	69	-72
2	14	4	14	56	-75
1	14	4	20	56	-64
1	13	4	20	46	-57
1	10	4	20	40	-50
1	6	4	12	24	-50
1	2	4	4	8	-50
3	2	4	6	8	-25
3	2	4	6	8	-25
0	2	4	8	8	0
0	1	3	3	3	0
1	1	2	2	3	-33
2	1	1	1	3	-67
3	1	0	0	N/A	N/A

In the initial stages of the generalization, the complexity measures provide a relatively inaccurate estimate of the potential for interestingness of the remaining portions of the concept hierarchies. This is because they consider only the complexity of the remaining nodes in the hierarchies that can be reached by generalization, but not the proportioning of the concept values from the hierarchies in the relation being generalized. For example, consider a single concept value for each of two attributes in a relation, with one having many more instances in the relation than the other. The importance of the ancestor nodes of the more frequently occurring concept is not assigned any more or less weight than the ancestor nodes of the less frequently occurring value. One reason these strategies perform better near the end of the generalization process is that there are fewer multiple occurrences of the same concept values and the complexity measures provide

better guidance towards interesting concepts. Additionally, since the complexity of the ANY node in a hierarchy is zero, an attribute will not be generalized to the ANY concept until there are no non-ANY generalizations to perform. This consistently seems to be an important factor in determining the ability of a strategy to produce interesting relations.

Figure 5.15 and 5.16 show the $\%d_{1a}$ versus tuple range for relation with 100 or fewer tuples for strategies S4 and S5 respectively. The strategies display reasonable

Figure 5.15: $\%d_{1a}$ versus Tuple Range for Strategy S4

performance over the entire range and display a decreasing trend in the negativity of $\%d_{1a}$ as the number of tuples reduces.

Figure 5.16: %d_{1a} versus Tuple Range for Strategy S5

5.8 Comparison of the Attribute Selection Strategies

Table 5.10 presents a summary of the %d_{1a} values calculated for relations with 100 or fewer tuples for the tasks run against the SaskTel database. In this table, the strategies are ranked by %d_{1a} values for 10 tuple ranges. Similar ranked %d_{1a} data is given for the tasks run against the NSERC database in Table 5.11. Table 5.12 presents a ranked summary of the %d_{2a} values calculated for relations with 100 or fewer tuples for the SaskTel tasks. Similar data is given for the NSERC tasks in Table 5.13. After ranking the strategies, the %d_{1a} and %d_{2a} values were evaluated to determine the statistical significance of the differences in magnitude. Details about the statistical analysis of the summaries is contained in Appendix B.

The ranked summaries show that strategies S4, S9 and S10 are the most effective in producing interesting relations. However, no single strategy provides the best

Table 5.10: Strategies Ranked by $\%d_{1a}$ for SaskTel Tasks

Range	Rank = 1		Rank = 2		Rank = 3		Rank = 4		Rank = 5		Rank = 6		Rank = 7		Rank = 8	
	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$
1-10	S4	-10.7	S1	-21.6	S5	-24.2	S2	-36.9	S9	-57.5	S10	-61.8	S3	-65.4	S8	-69.0
11-20	S4	-20.0	S1	-32.9	S5	-34.1	S9	-34.4	S10	-36.6	S3	-50.2	S2	-51.1	S8	-77.8
21-30	S4	-25.0	S1	-35.5	S9	-36.1	S5	-38.9	S10	-40.6	S2	-52.8	S3	-53.3	S8	-75.8
31-40	S4	-26.7	S9	-32.9	S1	-33.9	S10	-37.7	S5	-42.5	S3	-46.5	S2	-49.5	S8	-76.5
41-50	S9	-26.0	S10	-28.8	S4	-33.1	S1	-39.7	S3	-42.5	S5	-48.4	S2	-52.0	S8	-76.3
51-60	S9	-20.5	S10	24.4	S4	-39.6	S1	-45.0	S3	-50.0	S5	-53.8	S2	-56.6	S8	-74.3
61-70	S9	-15.3	S10	-19.2	S3	-32	S4	-35.3	S1	-39.7	S5	-48.1	S2	-50.4	S8	-70.8
71-80	S9	-26.6	S10	-34.1	S4	-34.4	S1	-41.3	S3	-43.6	S2	-49.9	S5	-53.4	S8	-72.3
81-90	S9	-13.8	S10	-16.4	S4	-30.7	S3	-38.5	S1	-46.2	S2	-49.1	S5	-52.6	S8	-67.7
91-100	S9	-10.6	S10	-17.0	S3	-31.0	S4	-44.6	S2	-52.6	S1	-53.7	S5	-54.6	S8	-67.2

Table 5.11: Strategies Ranked by $\%d_{1a}$ for NSERC Tasks

Range	Rank = 1		Rank = 2		Rank = 3		Rank = 4		Rank = 5		Rank = 6		Rank = 7		Rank = 8	
	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$
1-10	S4	-12.9	S5	-14.7	S1	-28.6	S2	-36.6	S9	-47.0	S10	-48.7	S3	-58	S8	-69.4
11-20	S4	-11.1	S5	-13.7	S9	-36.9	S10	-39.0	S3	-47.1	S1	-50.1	S2	-54.1	S8	-81.1
21-30	S4	-12.1	S5	-20.5	S9	-24.2	S10	-25.8	S3	-37.0	S1	-49.9	S2	-53.6	S8	-81.8
31-40	S4	-15.9	S9	-20.9	S10	-22.6	S5	-26.0	S3	-30.4	S1	-43.8	S2	-56.2	S3	-78.5
41-50	S4	-23.3	S9	-23.8	S1	-24.5	S5	-33.3	S3	-33.8	S1	-51.6	S2	-52.5	S8	-76.3
51-60	S9	-16.7	S10	-17.0	S3	-27.1	S4	-28.4	S5	-36.9	S1	-49.7	S2	-51.7	S8	-76.9
61-70	S10	-16.3	S9	-17.2	S3	-22.0	S4	-29.2	S5	-40.4	S2	-52.4	S1	-53.2	S8	-75.8
71-80	S10	-15.2	S9	-15.3	S3	-20.8	S4	-29.0	S5	-40.4	S2	-56.0	S1	-56.5	S8	-76.6
81-90	S9	-15.4	S10	-15.4	S3	-20.0	S4	-32.6	S5	-40.1	S2	-46.4	S1	-55.7	S8	-75.4
91-100	S9	-15.8	S10	-16.7	S3	-17.7	S4	-21.4	S5	-29.5	S2	-51.2	S1	-52.0	S8	-76.8

Table 5.12: Strategies Ranked by $\%d_{2a}$ for SaskTel Tasks

Range	Rank = 1		Rank = 2		Rank = 3		Rank = 4		Rank = 5		Rank = 6		Rank = 7		Rank = 8	
	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$
1-10	S4	-10.2	S1	-16.5	S5	-18.2	S2	-30.3	S9	-59.3	S10	-60.9	S3	-63.8	S8	-66.7
11-20	S4	-14.1	S1	-23.4	S5	-24.1	S10	-33.7	S9	-33.7	S2	-43.8	S3	-46.1	S8	-75.1
21-30	S4	-17.5	S1	-25.6	S5	-28.2	S9	-35.0	S10	-36.4	S2	-46.6	S3	-49.4	S8	-75.8
31-40	S4	-18.7	S1	-23.5	S9	-31.5	S5	-31.8	S10	-33.3	S2	-41.2	S3	-41.3	S8	-73.8
41-50	S4	23.5	S9	-24.0	S10	-24.8	S1	-30.1	S3	-36.4	S5	-38.0	S2	-45.1	S8	-72.2
51-60	S10	-22.4	S9	-22.8	S4	-31.5	S1	-36.7	S5	-44.6	S3	-46.6	S2	-49.2	S8	-71.8
61-70	S9	-16.7	S10	-17.0	S4	-27.3	S3	-28.2	S1	-30.8	S5	-37.7	S2	-40.8	S8	-68.9
71-80	S9	-25.1	S4	-26.4	S10	-29.4	S1	-33.3	S3	-38.7	S2	-42.0	S5	-43.7	S8	-68.3
81-90	S10	-14.1	S9	-14.2	S4	-22.5	S3	-34.9	S1	-38.0	S2	-41.8	S5	-43.6	S8	-66.0
91-100	S9	-12.1	S10	-14.4	S3	-28.5	S4	-36.8	S1	-43.3	S5	-43.6	S2	-46.2	S8	-65.8

Table 5.13: Strategies Ranked by $\%d_{2a}$ for NSERC Tasks

Range	Rank = 1		Rank = 2		Rank = 3		Rank = 4		Rank = 5		Rank = 6		Rank = 7		Rank = 8	
	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$
1-10	S4	-15.7	S5	-17.0	S1	-35.2	S2	-41.2	S9	-50.0	S10	-50.4	S3	-61.1	S8	-70.2
11-20	S4	-12.0	S5	-14.0	S9	-38.7	S10	-39.8	S1	-49.9	S3	-50.2	S2	-56.2	S8	80.6
21-30	S4	-15.0	S5	-20.2	S9	-25.8	S10	-26.2	S3	-39.4	S1	-48.3	S2	-57.4	S8	-80.4
31-40	S4	-16.4	S9	-21.6	S10	-22.0	S5	-24.2	S3	-31.6	S1	-43.8	S2	-59.8	S8	-76.9
41-50	S4	-22.9	S10	-23.4	S9	-24.0	S5	-31.5	S3	-34.8	S1	-49.2	S2	-56.7	S8	-75.1
51-60	S10	-15.6	S9	-16.3	S4	-27.6	S3	-29.3	S5	-34.1	S1	-46.6	S2	-56.3	S8	-75.4
61-70	S10	-14.6	S9	-16.4	S3	-23.1	S4	-26.3	S5	-36.5	S1	-49.2	S2	-57.2	S8	-75.1
71-80	S10	-15.5	S9	-16.4	S3	-23.0	S4	-27.7	S5	-36.3	S1	-52.4	S2	-60.3	S8	-76.1
81-90	S10	-14.7	S9	-16.9	S3	-21.6	S4	-30.4	S5	-35.7	S2	-52	S1	-52.3	S8	-76.1
91-100	S9	-17.2	S10	-17.7	S3	-18.7	S4	-19.3	S5	-26.4	S1	-49.2	S2	-55.8	S8	-77.2

performance over the entire range of interest. For tasks run against both the SaskTel and NSERC databases, strategy S4 yields significantly lower $\%d_{1a}$ and $\%d_{2a}$ values for relations with 1 to 40 tuples than any of the other strategies tested. For tasks run against both the NSERC and SaskTel databases, either strategy S9 or strategy S10 produced the lowest $\%d_{1a}$ values and $\%d_{2a}$ values for relations with 51 to 100 tuples.

The tuple range 41 to 50 seems to be a transition range between strategy S4 and strategies S9 and S10. For the SaskTel tasks, Strategy S9 and strategy S10 appear to be significantly more effective in this range than strategy S4 when ranked by $\%d_{1a}$. Strategy S4 has a higher rank than strategies S9 and S10 for the SaskTel tasks when ranked by $\%d_{2a}$. However, the differences in $\%d_{1a}$ values or $\%d_{2a}$ values are not significant. Strategy S4 appears to be the most effective in this range for the NSERC tasks, but its $\%d_{1a}$ and $\%d_{2a}$ values are not significantly higher than those for strategy 9 or strategy 10.

As previously discussed, strategies S9 and S10 have similar behavior. For the NSERC tasks, strategy S9 has lower $\%d_{1a}$ values than strategy S10 for 7 of the 10 ranges. However, the $\%d_{1a}$ values for the two strategies are not significantly different. For the SaskTel tasks, strategy S9 has lower $\%d_{1a}$ values than strategy S10 for all ranges, with the difference being significant for 9 of the 10 ranges. In part, the slightly better performance of $\%d_{1a}$ values for strategy S9 occurs because S9 makes attribute selections based on I_1 , the same measure used to calculate $\%d_{1a}$. When differences in $\%d_{2a}$ are considered, strategy S10 has lower $\%d_{2a}$ values than strategy S9 in 5 of 10 ranges for the SaskTel tasks, with no significant difference in the values. For the SaskTel tasks, strategy S10 has a higher $\%d_{2a}$ value than strategy S9 in three of ten ranges, with the difference in $\%d_{2a}$

values being significant in only 1 of the ten ranges. As a result of this somewhat contradictory evidence, further testing on concept hierarchies with greater variation is required to select the best alternative between these strategies.

Strategy S4 is the most effective for small relations and since its time complexity is $O(m)$, compared to $O(m^2 n_i n_o)$ for strategies S9 and S10, it is the best choice of the original strategies tested.

5.9 Analysis of Modified Strategies S11 and S12

As discussed in the previous section, the experimental data indicates that no strategy performed well over the entire 100 tuple range being considered. Strategy S4 performed well in relations with 1 to 40 tuples and strategies S9 and S10 performed well in relations with 51 to 100 tuples.

One reason that has been identified for the poor performance of strategies S9 and S10 for smaller relations is a tendency to generalize an attribute to the ANY concept relatively early in the generalization process. This illustrates that the short-sighted approach of one-step lookahead strategies can result in the selection of attributes near the start of the generalization process that may not lead to the best results. To see whether the performance of strategies S9 and S10 could be improved for smaller relations, the strategies were modified so that a generalization to the ANY concept was not permitted until no other type of generalization was available. For reference, the modified strategy based on strategy S9 is called strategy S11 and the modified strategy based on strategy S10 is referred to as strategy S12. Strategies S11 and S12 were tested using the same tasks as the original strategies.

Figure 5.17 shows a plot of the I_I values of relations produced in task T27 when attribute selection was based on the two modified strategies S11 and S12. Table 5.14 provides the task generalization information for S11 and Table 5.15 for S12.

Figure 5.17 : Strategies S11 and S12 I_I versus Number of Tuples for Task T27

For the sample task, these strategies provide good performance over the entire task, with an average $\%d_{Ia}$ value of -15% for strategy S11 and -20% for strategy S12. For both strategies, the I_I value equals the $I_{I\max_n}$ value for 10 of the 19 relations produced. For 8 of the 9 remaining relations, it is impossible for the strategies to produce the $I_{I\max_n}$ values because some attributes are either not yet generalized, or maximally generalized.

Table 5.14 : Strategy S11 Generalization Information for Task T27

index	n	m _a	l ₁	l ₁ max _n	%d ₁
N/A	59	4	0	N/A	N/A
0	59	4	59	177	-67
3	59	4	118	177	-33
1	59	4	148	177	-16
2	57	4	168	196	-14
1	57	4	180	196	-8
1	57	4	196	196	0
3	57	4	196	196	0
2	55	4	191	191	0
2	52	4	208	208	0
1	46	4	184	184	0
1	36	4	144	144	0
2	14	4	56	56	0
2	8	4	32	32	0
2	4	4	16	16	0
2	2	4	8	8	0
1	2	3	6	8	-25
2	2	2	4	8	-50
3	2	1	2	8	-75
0	1	0	0	N/A	N/A

Table 5.15 : Strategy S12 Generalization Information for Task T27

index	n	m _a	l ₁	l ₁ max _n	%d ₁
N/A	59	4	0	N/A	N/A
2	57	4	25	196	-87
1	57	4	54	196	-72
3	57	4	111	196	-43
0	57	4	168	196	-14
1	57	4	180	196	-8
1	57	4	196	196	0
3	57	4	196	196	0
2	55	4	191	191	0
2	52	4	208	208	0
1	46	4	184	184	0
1	36	4	144	144	0
2	14	4	56	56	0
2	8	4	32	32	0
2	4	4	16	16	0
2	2	4	8	8	0
3	2	3	6	8	-25
1	2	2	4	8	-50
2	2	1	2	8	-75
0	1	0	0	N/A	N/A

Figures 5.18 is a plot of $\%d_{la}$ values versus tuple range for relations with 100 tuples or fewer produced by strategy S11. Figures 5.19 is a plot of $\%d_{la}$ values versus

Figure 5.18: $\%d_{la}$ versus Tuple Range for Strategy S11

Figure 5.19: $\%d_{la}$ versus Tuple Range for Strategy S12

tuple range for relations with 100 tuples or fewer produced by strategy S12. In general, the modified strategies perform well for relations with 21 to 100 tuples. However, Figure 5.18 and Figure 5.19 show there is still an increasing trend in the negativity of $\%d_{1a}$ for relations with 20 or fewer tuples.

The ranked results comparing the $\%d_{1a}$ for the new strategies to those for strategies S4, S9 and S10 are presented in Table 5.16 for the SaskTel tasks and Table 5.17 for the NSERC tasks. Similar information is provided comparing $\%d_{2a}$ in Table 5.18 for the SaskTel tasks and Table 5.19 for the NSERC tasks.

The modified strategies S11 and S12 provide a significant improvement over the original strategies S9 and S10 for all tuple ranges. However, strategy S4 still provides significantly lower $\%d_{a1}$ values for relations with 20 or fewer tuples in the NSERC tasks and 10 or fewer tuples in the SaskTel tasks. One possible reason for this is that strategy S4 examines the entire remaining concept hierarchies in selecting an attribute to

Table 5.16: Comparison of $\%d_{1a}$ of Modified Strategies for SaskTel Tasks

Range	Rank = 1		Rank = 2		Rank = 3		Rank = 4		Rank = 5	
	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$
1-10	S4	-10.7	S12	-19.5	S11	-19.6	S9	-57.5	S10	-61.8
11-20	S12	-9.2	S11	-10.0	S4	-14.1	S9	-59.3	S10	-60.9
21-30	S11	-7.9	S12	-9.0	S4	-17.5	S9	-35.0	S10	-36.4
31-40	S11	-2.5	S12	-2.9	S4	-18.7	S9	-31.5	S10	-33.3
41-50	S11	-3.9	S12	-5.0	S4	23.5	S9	-24.0	S10	-24.8
51-60	S11	-13.3	S12	-15.5	S10	-22.4	S9	-22.8	S4	-31.5
61-70	S11	-5.0	S12	-7.0	S9	-16.7	S10	-17.0	S4	-27.3
71-80	S11	-7.0	S12	-8.7	S9	-25.1	S4	-26.4	S10	-29.4
81-90	S11	-4.1	S12	-6.3	S10	-14.1	S9	-14.2	S4	-22.5
91-100	S11	-8.4	S12	-13.4	S9	-12.1	S10	-14.4	S4	-36.8

Table 5.17: Comparison of $\%d_{1a}$ of Modified Strategies for NSERC Tasks

Range	Rank = 1		Rank = 2		Rank = 3		Rank = 4		Rank = 5	
	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$	S	$\%d_{1a}$
1-10	S4	-15.7	S11	-35.7	S12	-35.8	S9	-50.0	S10	-50.4
11-20	S4	-11.1	S11	-24.4	S12	-24.8	S9	-38.7	S10	-39.8
21-30	S11	-12.1	S12	12.8	S4	-15.0	S9	-25.8	S10	-26.2
31-40	S12	-12.8	S11	-13.1	S4	-16.4	S9	-21.6	S10	-22.0
41-50	S11	-15.3	S12	-15.4	S4	-22.9	S10	-23.4	S9	-24.0
51-60	S12	-10.9	S11	-11.2	S10	-15.6	S9	-16.3	S4	-27.6
61-70	S11	-7.9	S12	-8.0	S10	-14.6	S9	-16.4	S4	-26.3
71-80	S11	-13.8	S12	-14.7	S10	-15.5	S9	-16.4	S4	-27.7
81-90	S11	-7.7	S12	-7.8	S10	-14.7	S9	-16.9	S4	-30.4
91-100	S11	-10.4	S12	-10.4	S9	-17.2	S10	-17.7	S4	-19.3

Table 5.18: Comparison of $\%d_{2a}$ of Modified Strategies for SaskTel Tasks

Range	Rank = 1		Rank = 2		Rank = 3		Rank = 4		Rank = 5	
	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$
1-10	S4	-10.2	S12	-25.4	S11	-27.5	S9	-59.3	S10	-60.9
11-20	S12	-11.7	S11	-13.9	S4	-14.1	S10	-33.7	S9	-33.7
21-30	S12	-11.3	S11	-11.5	S4	-17.5	S9	-35.0	S10	-36.4
31-40	S11	-5.4	S12	-5.5	S4	-18.7	S9	-31.5	S10	-33.3
41-50	S11	-6.6	S12	-6.8	S4	23.5	S9	-24.0	S10	-24.8
51-60	S12	-16.4	S11	-17.5	S10	-22.4	S9	-22.8	S4	-31.5
61-70	S11	-7.2	S12	-7.4	S9	-16.7	S10	-17.0	S4	-27.3
71-80	S12	-9.4	S11	-9.5	S9	-25.1	S4	-26.4	S10	-29.4
81-90	S11	-6.1	S12	-6.6	S10	-14.1	S9	-14.2	S4	-22.5
91-100	S11	-10.3	S12	-12.0	S9	-12.1	S10	-14.4	S4	-36.8

Table 5.19: Comparison of $\%d_{2a}$ of Modified Strategies for NSERC Tasks

Range	Rank = 1		Rank = 2		Rank = 3		Rank = 4		Rank = 5	
	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$	S	$\%d_{2a}$
1-10	S4	-15.7	S12	-37.6	S11	-39.2	S9	-50.0	S10	-50.4
11-20	S4	-12.0	S12	-25.5	S11	-26.5	S9	-38.7	S10	-39.8
21-30	S12	-12.9	S11	-13.6	S4	-15.0	S9	-25.8	S10	-26.2
31-40	S12	-12.9	S11	-14.4	S4	-16.4	S9	-21.6	S10	-22.0
41-50	S12	-14.7	S11	-15.5	S4	-22.9	S10	-23.4	S9	-24.0
51-60	S12	-9.7	S11	-10.8	S10	-15.6	S9	-16.3	S4	-27.6
61-70	S12	-7.7	S11	-8.3	S10	-14.6	S9	-16.4	S4	-26.3
71-80	S12	-14.7	S11	-15.5	S10	-15.5	S9	-16.4	S4	-27.7
81-90	S12	-7.8	S11	-8.8	S10	-14.7	S9	-16.9	S4	-30.4
91-100	S12	-10.7	S11	-11.7	S9	-17.2	S10	-17.7	S4	-19.3

generalize. Strategies S11 and S12 only look ahead one step in selecting an attribute. It is difficult to recover from a poor selection made near the end of the generalization process. Given that strategy S4 is more efficient and that it is more effective for small relations, it remains the best alternative for an attribute selection strategy.

CHAPTER 6

CONCLUSIONS

This thesis describes an experimental study of the performance of attribute selection strategies for use in attribute-oriented generalization. The goal of the research was to replace the user-defined attribute threshold currently used in attribute-oriented generalization, with a strategy for selecting the next attribute to generalize that efficiently and effectively guides the user towards interesting results.

Section 6.1 of this chapter provides a brief summary and discussion of the experimental results that were obtained and outlines the main conclusions. Section 6.2 describes areas where future research would be valuable.

6.1 Attribute Selection for Generalization in AOG

Several attribute selection strategies suggested in the literature (Cai, Cercone and Han, 1991), (Han, Cai and Cercone, 1992), (Carter, Hamilton and Cercone, 1994) were experimentally tested in 638 discovery tasks performed on two different databases. The effectiveness of the strategies was evaluated using interestingness measures that consider the structure of the domain-expert defined concept hierarchies that guide the generalization process and the characteristics of the information specific to a discovery task. Six new strategies based on these interestingness measures were also examined.

The strategy based on selecting the attribute for generalization with the greatest complexity measure M_I proved to be significantly more effective than the other strategies tested for relations with 1 to 20 tuples. The success of this strategy can be attributed to two factors. First, the strategy delays the generalization of attributes to the ANY concept until the end of the generalization process, ensuring that all attributes are active for as long as possible. Strategies that allowed earlier generalization of attributes to the ANY concept were found to have poor performance after attributes became maximally generalized. Secondly, the strategy examines the potential for interestingness of the entire remaining hierarchies. Strategies using a single step lookahead were found to be more effective for large final relations but less effective for small final relations. It is more difficult to recover from a poor attribute selection near the end of the generalization process than the start of the process. For relations with 21 to 100 tuples, the single step lookahead strategies based selecting an attribute for generalization which produces the greatest I_1 or I_2 are the most effective, if they are modified to prevent the early generalization of attributes to the ANY concept.

The strategy based on selecting the attribute for generalization with the fewest distinct values, was easily the least effective of the strategies tested because it generalizes and eliminates attributes one at a time. This leads to the conclusion that all attributes should be generalized at least once before any attribute selection strategy is applied. This also reduces the possibility that the user will be presented with a mix of generalized and ungeneralized information.

Since the strategy based on the complexity measure M_I is most effective for small relations that are understandable to the user and it is more efficient than the strategies

based on I_1 or I_2 , it is recommended as the most suitable attribute selection strategy to replace the user-defined attribute threshold.

6.2 Areas For Future Research

One area for future research is a more extensive investigation of differences between the strategies based on I_1 and I_2 . In this work, there was no strong evidence to suggest that they produce different results. However, if the strategies were applied to concept hierarchies that differed more radically from balanced trees, a clearer indication of the differences may be obtained.

The one-step lookahead strategies do not perform as well for small relations as they do for larger relations. A possible area of future research is the testing multi-step lookahead strategies. For example, a lookahead strategy might examine the interestingness of the next two generalizations and select the first attribute of the pair of attributes whose generalization produced relations with the best average interestingness.

Strategy S4 performed better than strategies S11 and S12 in the 1-20 tuple range and strategies S11 and S12 performed better than strategy S4 in the 21-100 tuple range. A composite strategy combining strategy S4 with strategy S11 or strategy S12 would be worth investigating. In such a composite strategy, strategy S11 or S12 would be used to select the next attribute to generalize when a relation had greater than 20 tuples. When the number of tuples in the relations dropped below 20, strategy S4 would select the attributes for the remainder of the generalization process.

The interestingness measures I_1 and I_2 consider the structure of the domain specific concept hierarchies and the characteristics of the task specific data retrieved from

the database. However, they do not consider the variation of the data retrieved for a task from expected or normative values. One other promising area of future research is the investigation of modifications to the interestingness measures I_1 and I_2 so that they consider the variation of the data from expected values.

In general, Section 2.4.2 describes a variety of methods for specifying what is interesting about discovered knowledge. The investigation of different measures of interestingness is an important research area. This might include the development of new interestingness measures or the adaptation of existing measures to different domains (Hamilton, Hilderman and Cercone, 1996).

REFERENCES

- Agrawal, R., Imielinski, T. and Swami, A. (1993). "Mining Association Rules between Sets of Items in Large Databases," *Proceedings of the 1993 International Conference on Management of Data (SIGMOD '93)*, 207-216.
- Anand, S.S., Bell, D.A. and Hughes, J.G. (1995). "The Role of Domain Knowledge in Data Mining," *Proceedings of the 4th International Conference on Information and Knowledge Management (ACM CIKM '95)*, 37-43.
- Barber, D.B. and Hamilton, H.J. (1996). "Attribute Selection Strategies for Attribute-Oriented Generalization," *Proceedings of the Canadian AI Conference (AI'96)*, 429-441.
- Bhandari, L. (1993). "Attribute Focusing: Machine-Assisted Knowledge Discovery Applied to Software Production Process Control," *Knowledge Discovery in Databases: Papers from the 1993 Workshop*, Technical Report WS-93-02, AAAI Press, Menlo Park, CA., 61-69.
- Cai, Y., Cercone, N. and Han, J. (1989). "Learning Characteristic Rules from Relational Databases," *Proceedings of the International Symposium on Computational Intelligence*, Milano, Italy, 187-196.
- Cai, Y., Cercone, N. and Han, J. (1991). "Knowledge Discovery in Databases: An Attribute-Oriented Approach," in: Piatetsky-Shapiro, G. and Frawley, W.J., Eds., *Knowledge Discovery in Databases*, AAAI/MIT Press, Menlo Park, CA., 213-228.
- Carbonell, J.G., Michalski, R.S. and Mitchell, T.M. (1983). "An Overview of Machine Learning," in: Michalski, R.S., Carbonell, J.G. and Mitchell, T.M., Eds., *Machine Learning: An Artificial Intelligence Approach*, Tioga Pub. Co., Palo Alta, CA., 4-23.
- Carter, C.L. (1994). "Efficient Attribute-oriented Generalization for Knowledge Discovery From Large Databases," M. Sc. Thesis, 1994.
- Carter, C.L. and Hamilton, H.J. (1995a). "Fast Incremental Generalization and Regeneralization for Knowledge Discovery from Databases," *Proceedings of the Eighth Florida Artificial Intelligence Research Symposium (FLAIRS-95)*, Melbourne, FL., 319-323.
- Carter, C.L. and Hamilton, H.J. (1995b). "A Fast On-Line Generalization Algorithm for Knowledge Discovery," *Applied Mathematics Letters*, 8:2, 5-11.
- Carter, C.L., Hamilton, H.J. and Cercone, N. (1994). *The Software Architecture of DBLEARN*, Technical Report CS-94-04, University of Regina.

Cheung, D.W., Fu, A.W. and Han, J. (1994). "Knowledge Discovery in Databases: A Rule-Based Attribute-Oriented Approach," in: Ras, Z.W. and Zemankova, M., Eds., *Lecture Notes in Artificial Intelligence, 8th International Symposium, ISMIS '94*, Charlotte, N.C., 164-173.

Dietterich, T.G. and Michalski, R.S. (1983). "A Comparative Review of Selected Methods for Learning from Examples," in: Michalski, R.S., Carbonell, J.G. and Mitchell, T.M., Eds., *Machine Learning: An Artificial Intelligence Approach*, Tioga Pub. Co., Palo Alto, CA., 41-81.

Fayyad, U.M., Piatetsky-Shapiro, G. and Smyth, P. (1996). "From Data Mining to Knowledge Discovery: An Overview," in: Fayyad, U.M., Piatetsky-Shapiro, G. and Smyth, P., Eds., *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, Menlo Park, CA., 1-34.

Fayyad, U.M., Weir, N. and Djorgovski, A. (1993). "Automated Analysis of a Large-Scale Sky Survey: The SKYCAT System," *Knowledge Discovery in Databases: Papers from the 1993 Workshop*, Technical Report WS-93-02, AAAI Press, Menlo Park, CA., 1-13.

Fayyad, U.M., Djorgovski, A. and Weir, N. (1996). "Automating the Analysis and Cataloging of Sky Surveys," in: Fayyad, U.M., Piatetsky-Shapiro, G. and Smyth, P., Eds., *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, Menlo Park, CA., 473-493.

Feldman, R. and Dagan, I. (1995). "Knowledge Discovery in Textural Databases (KDT)," *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, Montreal, Quebec, Canada, 112-117.

Frawley, W.J., Piatetsky-Shapiro, G. and Matheus, C.J. (1991). "Knowledge Discovery in Databases: An Overview," in: Piatetsky-Shapiro, G. and Frawley, W.J., Eds., *Knowledge Discovery in Databases*, AAAI/MIT Press, Menlo Park, CA., 1-27.

Hamilton, H.J. and Fudger, D.F. (1995). "Estimating DBLEARN's Potential for Knowledge Discovery in Databases," *Computational Intelligence*, 11:2, 1-18.

Hamilton, H.J., Hilderman, R.J. and Cercone, N. (1996). "Attribute-Oriented Induction Using Domain Generalization Graphs," *Proceedings of the Eight IEEE International Conference on Tools with Artificial Intelligence (ICTAI-96)*, Toulouse, France, 271-274.

Han, J., Cai, Y. and Cercone, N. (1991). "Concept-Based Data Classification in Relational Databases," *Workshop Notes of the 1991 AAAI Workshop on Knowledge Discovery in Databases*, Anaheim, CA., 77-94.

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

Han, J., Cai, Y. and Cercone, N. (1993). "Data-Driven Discovery of Quantitative Rules in Relational Databases," *IEEE Transactions on Knowledge and Data Engineering*, 5:1, 29-39.

Han, J., Cai, Y., Cercone, N. and Huang, Y. (1995). "Discovery of Data Evolution Regularities in Large Databases," *Journal of Computer and Software Engineering* (Special Issue on Methodologies and Tools for Intelligent Information Systems), 3:1, 41-69.

Han, J., Nishio, S. and Kawano, H. (1993). "Knowledge Discovery in Object-Oriented and Active Databases," *Proceedings of the 1993 International Conference on Building and Sharing of Very Large Scale Knowledge Bases*, Tokyo, Japan, 205-214.

Holsheimer, M. and Siebes, A. (1994). "Data Mining: The Search for Knowledge in Databases," *Technical Report CS-R9406*, Center for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands.

Hwang, H. and Fu, W. (1995). "Efficient Algorithms for Attribute-Oriented Induction," *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, Montreal, Quebec, Canada, 168-173.

Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H. and Verkamo, A.I. (1994). "Finding Interesting Rules from Large Sets of Discovered Association Rules," in: Adams, N.R., Bhargava, B.K. and Yesha, Y., Eds., *Third International Conference on Information and Knowledge Management*, ACM Press, Gaitersburg, Md., 401-407.

Klösgen, W. (1992). "Problems for Knowledge Discovery in Databases and Their Treatment in the Statistics Interpreter Explora," *International Journal of Intelligent Systems*, Vol. 7, 1-13.

Major, J.A. and Mangano, J.J. (1993). "Selecting Among Rules Induced from a Hurricane Database," *Knowledge Discovery in Databases: Papers from the 1993 Workshop*, Technical Report WS-93-02, AAAI Press, Menlo Park, CA., 28-41.

Matheus, C.J., Chan, P.K. and Piatetsky-Shapiro, G. (1993). "Systems for Knowledge Discovery in Databases," *IEEE Transactions on Knowledge and Data Engineering*, 5:6, 903-913.

Matheus, C.J. and Piatetsky-Shapiro, G. (1995). "Selecting and Reporting What is Interesting: The KEFIR Application to Healthcare Data," in: *Advances in Knowledge*

Discovery and Data Mining, Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R., Eds., AAAI/MIT Press, Menlo Park, CA., 401-419.

Michalski, R.S. (1983). "A Theory and Methodology of Inductive Learning," in: Michalski, R.S., Carbonell, J.G. and Mitchell, T.M., Eds., *Machine Learning: An Artificial Intelligence Approach*, Tioga Pub. Co., Palo Alta, CA., 83-134.

Nishio, S., Kawano, H. and Han, J. (1993). "Knowledge Discovery in Object-Oriented Databases: The First step," *Knowledge Discovery in Databases: Papers from the 1993 Workshop*, Technical Report WS-93-02, AAAI Press, Menlo Park, CA., 299-213.

Piatetsky-Shapiro, G. (1991). "Discovery, Analysis and Presentation of Strong Rules," in: Piatetsky-Shapiro, G. and Frawley, W.J., Eds., *Knowledge Discovery in Databases*, AAAI/MIT Press, Menlo Park, CA., 229-248.

Rivera, C.B. and Carter, C.L. (1995). *A Tutorial Guide to DB-Discover, Version 2.0*, Technical Report CS-95-05, University of Regina.

Shan, N., Hamilton, H.J. and Cerccone, N. (1995). "GRG: Knowledge Discovery Using Information Generalization, Information Reduction and Rule Generation," 7th *IEEE International Conference on Tools with Artificial Intelligence*, Washington, D.C., 372-379.

Silberschatz, A. and Tuzhilin, A. (1995). "On Subjective Measures of Interestingness in Knowledge Discovery," *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, Montreal, Quebec, Canada, 275-281.

Smyth, P. and Goodman, R.M. (1991). "Rule Induction using Information Theory," in: Piatetsky-Shapiro, G. and Frawley, W.J., Eds., *Knowledge Discovery in Databases*, AAAI/MIT Press, Menlo Park, CA., 159-176.

Smyth, P. and Goodman, R.M. (1992), "An Information Theoretic Approach to Rule Induction from Databases," *IEEE Transactions on Knowledge and Data Engineering*, 4:4, 301-316.

APPENDIX A

CHARTS OF $\%d_{2a}$ VERSUS TUPLE RANGE

Figure A.1: $\%d_{2a}$ versus Tuple Range for Strategy S1

Figure A.2: $\%d_{2a}$ versus Tuple Range for Strategy S2

Figure A.3: %d_{2a} versus Tuple Range for Strategy S3

Figure A.4: %d_{2a} versus Tuple Range for Strategy S4

Figure A.5: $\%d_{2a}$ versus Tuple Range for Strategy S5

Figure A.6: $\%d_{2a}$ versus Tuple Range for Strategy S8

Figure A.7: $\%d_{2a}$ versus Tuple Range for Strategy S9

Figure A.8: $\%d_{2a}$ versus Tuple Range for Strategy S10

Figure A.9: %d_{2a} versus Tuple Range for Strategy S11

Figure A.10: %d_{2a} versus Tuple Range for Strategy S12

APPENDIX B

STATISTICAL ANALYSIS OF EXPERIMENTAL RESULTS

In this appendix, the statistical analysis of the experimental test results is outlined. Section A.1 provides a summary of the calculated means and standard deviations of the $\%d_1$ and $\%d_2$ values that were used to compare the various strategies. Section A.2 describes the statistical analysis of the ranked mean $\%d_1$ and mean $\%d_2$ values and provides a summary of the statistical information.

A.1 Summary of Means and Standard Deviations for $\%d_1$ and $\%d_2$ Values

The percent differences $\%d_1$ and $\%d_2$, as defined in Section 5.4, were calculated for all relations generated by all strategies for all tasks. The mean $\%d_1$ and $\%d_2$ values were calculated for all relations with 100 or fewer tuples, in ten tuple increments for each of the strategies using the equations

$$\%d_{1a} = \sum \%d_1 / n \quad \dots (B.1)$$

$$\%d_{2a} = \sum \%d_2 / n \quad \dots (B.2)$$

where n is the number of relations with a size within the specified tuple range. The corresponding standard deviations were determined using the equations

$$s_1 = \left(\sum_{i=1}^n (\%d_{1i} - \%d_{1a})^2 / (n-1) \right)^{1/2} \quad \dots (B.3)$$

$$s_2 = \left(\sum_{i=1}^n (\%d_{2i} - \%d_{2a})^2 / (n-1) \right)^{1/2} \quad \dots (B.4)$$

where $\%d_{1i}$ is the $\%d_1$ value of the i^{th} relation with a size within the tuple range and $\%d_{2i}$ is the $\%d_2$ value of the i^{th} relation with a size within the tuple range. The calculated mean and standard deviation values are summarized in Tables B.1 through to B.10 for the SaskTel tasks and in Tables B.11 through B.20 for the NSERC tasks. Although the real numbers in these tables are shown with one decimal place of accuracy, the calculations were done using a electronic spreadsheet program with 15 decimal places of accuracy.

Table B.1: Tuple Range 1-10 $\%d_{1a}$ and $\%d_{2a}$ Values for SaskTel Tasks

Strategy	n	$\%d_{1a}$	s_1	$\%d_{2a}$	s_2
S1	1704	-21.6	19.5	-16.5	15.8
S2	1448	-36.9	26.8	-30.3	27.7
S3	871	-65.4	21.4	-63.8	24.2
S4	2729	-10.7	16.7	-10.2	13.0
S5	2386	-24.2	18.3	-18.2	14.2
S8	949	-69.0	16.9	-66.7	20.2
S9	1038	-57.5	21.9	-59.3	22.8
S10	860	-61.8	23.3	-60.9	25.2
S11	1947	-19.6	26.9	-27.5	30.7
S12	1952	-19.5	26.7	-25.4	29.1

Table B.2: Tuple Range 11-20 % d_{1a} and % d_{2a} Values for SaskTel Tasks

Strategy	n	% d_{1a}	s_1	% d_{2a}	s_2
S1	751	-32.9	21.7	-23.4	19.1
S2	1350	-51.1	25.8	-43.8	27.8
S3	587	-50.2	25.8	-46.1	25.9
S4	1144	-20.0	20.4	-14.1	15.5
S5	1351	-34.1	21.3	-24.1	17.5
S8	509	-77.8	18.3	-75.1	18.0
S9	694	-34.4	24.3	-33.7	24.0
S10	695	-36.6	25.3	-33.7	23.9
S11	724	-10.0	19.2	-13.9	20.8
S12	712	-9.2	18.8	-11.7	18.2

Table B.3: Tuple Range 21-30 % d_{1a} and % d_{2a} Values for SaskTel Tasks

Strategy	n	% d_{1a}	s_1	% d_{2a}	s_2
S1	438	-35.5	24.4	-25.6	22.1
S2	643	-52.8	23.4	-46.6	25.4
S3	370	-53.3	24.9	-49.4	26.4
S4	572	-25.0	23.9	-17.5	19.6
S5	726	-38.9	22.9	-28.2	20.0
S8	329	-75.8	17.0	-74.7	18.1
S9	465	-36.1	26.3	-35.0	25.5
S10	474	-40.6	25.6	-36.4	24.4
S11	335	-7.9	16.6	-11.5	19.2
S12	335	-9.0	18.5	-11.3	17.7

Table B.4: Tuple Range 31-40 % d_{1a} and % d_{2a} Values for SaskTel Tasks

Strategy	n	% d_{1a}	s_1	% d_{2a}	s_2
S1	434	-33.6	22.0	-23.5	19.5
S2	393	-49.5	22.2	-41.2	22.2
S3	341	-46.5	27.7	-41.3	27.4
S4	343	-26.7	23.5	-18.7	19.4
S5	462	-42.5	22.5	-31.8	20.2
S8	363	-76.5	19.4	-73.8	19.7
S9	529	-32.9	23.5	-31.5	23.2
S10	516	-37.7	25.2	-33.3	24.3
S11	345	-2.5	8.7	-5.4	10.2
S12	343	-2.9	10.1	-5.5	10.0

Table B.5: Tuple Range 41-50 % d_{1a} and % d_{2a} Values for SaskTel Tasks

Strategy	n	% d_{1a}	s_1	% d_{2a}	s_2
S1	369	-39.7	26.8	-30.1	24.8
S2	439	-52.0	26.0	-45.1	26.4
S3	522	-42.5	23.1	-36.4	22.2
S4	313	-33.1	29.1	-23.5	26.0
S5	381	-48.4	24.2	-38.0	23.3
S8	733	-76.3	17.7	-72.2	19.4
S9	504	-26.0	22.8	-24.0	20.6
S10	476	-28.8	25.6	-24.8	21.6
S11	428	-3.9	11.4	-6.6	13.1
S12	446	-5.0	13.6	-6.8	11.8

Table B.6: Tuple Range 51-60 % d_{1a} and % d_{2a} Values for SaskTel Tasks

Strategy	n	% d_{1a}	s_1	% d_{2a}	s_2
S1	344	-45.0	28.3	-36.7	26.8
S2	447	-56.6	23.5	-49.2	25.2
S3	419	-50.0	25.9	-46.6	27.3
S4	302	-39.6	29.0	-31.5	27.5
S5	275	-53.8	24.1	-44.6	23.3
S8	515	-74.3	21.7	-71.8	22.5
S9	369	-20.5	21.4	-22.8	22.9
S10	357	-24.4	23.8	-22.4	20.8
S11	305	-13.3	20.2	-17.5	23.5
S12	301	-15.5	23.4	-16.4	21.1

Table B.7: Tuple Range 61-70 % d_{1a} and % d_{2a} Values for SaskTel Tasks

Strategy	n	% d_{1a}	s_1	% d_{2a}	s_2
S1	212	-39.7	24.9	-30.8	23.5
S2	230	-50.4	20.6	-40.8	21.4
S3	211	-32.0	24.4	-28.2	23.3
S4	211	-35.3	26.9	-27.3	25.9
S5	245	-48.1	22.6	-37.7	22.8
S8	317	-70.8	25.4	-68.9	25.4
S9	318	-15.3	18.6	-16.7	19.4
S10	304	-19.2	20.9	-17.0	16.8
S11	260	-5.0	11.3	-7.2	12.7
S12	264	-7.0	15.5	-7.4	11.6

Table B.8: Tuple Range 71-80 $\%d_{1a}$ and $\%d_{2a}$ Values for SaskTel Tasks

Strategy	n	$\%d_{1a}$	s_1	$\%d_{2a}$	s_2
S1	160	-41.3	28.6	-33.3	27.5
S2	198	-49.9	24.1	-42.0	24.8
S3	575	-43.6	27.1	-38.7	25.9
S4	173	-34.4	30.1	-26.4	28.7
S5	177	-53.4	23.7	-43.7	24.2
S8	461	-72.3	20.5	-68.3	21.6
S9	520	-26.6	25.9	-25.1	23.8
S10	624	-34.1	27.3	-29.4	24.3
S11	293	-7.0	14.3	-9.5	15.7
S12	321	-8.7	16.2	-9.4	12.7

Table B.9: Tuple Range 81-90 $\%d_{1a}$ and $\%d_{2a}$ Values for SaskTel Tasks

Strategy	n	$\%d_{1a}$	s_1	$\%d_{2a}$	s_2
S1	150	-46.2	24.6	-38.0	24.3
S2	104	-49.1	19.3	-41.8	20.7
S3	209	-38.5	25.7	-34.9	24.3
S4	141	-30.7	27.0	-22.5	25.1
S5	156	-52.6	22.6	-43.6	23.1
S8	140	-67.7	28.9	-66.0	28.7
S9	288	-13.8	19.0	-14.2	17.9
S10	273	-16.4	20.9	-14.1	17.3
S11	252	-4.1	10.8	-6.1	11.7
S12	269	-6.3	15.4	-6.6	12.3

Table B.10: Tuple Range 91-100 $\%d_{1a}$ and $\%d_{2a}$ Values for SaskTel Tasks

Strategy	n	$\%d_{1a}$	s_1	$\%d_{2a}$	s_2
S1	191	-53.7	24.4	-43.3	23.3
S2	149	-52.6	24.0	-46.2	23.5
S3	335	-31.0	26.3	-28.5	25.5
S4	210	-44.6	31.0	-36.8	28.9
S5	181	-54.6	31.0	-43.6	28.6
S8	316	-67.2	17.8	-65.8	17.3
S9	260	-10.6	26.0	-12.1	26.7
S10	340	-17.0	21.7	-14.4	17.9
S11	205	-8.4	17.8	-10.3	19.1
S12	240	-13.4	22.3	-12.0	18.2

Table B.11: Tuple Range 1-10 % d_{1a} and % d_{2a} Values for NSERC Tasks

Strategy	n	% d_{1a}	s_1	% d_{2a}	s_2
S1	488	-28.6	30.8	-35.2	32.1
S2	441	-36.6	31.5	-41.2	31.8
S3	247	-58.0	18.7	-61.1	18.7
S4	267	-12.9	19.7	-15.7	18.3
S5	275	-14.7	22.0	-17.0	20.3
S8	293	-69.4	19.3	-70.2	19.6
S9	230	-47.0	22.7	-50.0	22.9
S10	228	-48.7	22.3	-50.4	22.5
S11	307	-35.7	26.6	-39.2	27.6
S12	307	-35.8	26.6	-37.6	27.3

Table B.12: Tuple Range 11-20 % d_{1a} and % d_{2a} Values for NSERC Tasks

Strategy	n	% d_{1a}	s_1	% d_{2a}	s_2
S1	198	-50.1	25.8	-49.9	25.8
S2	210	-54.1	26.1	-56.2	26.0
S3	195	-47.1	20.5	-50.2	21.1
S4	206	-11.1	22.6	-12.0	21.8
S5	217	-13.7	24.3	-14.0	22.5
S8	357	-81.1	16.0	-80.6	16.3
S9	251	-36.9	21.8	-38.7	23.3
S10	232	-39.0	22.0	-39.8	23.3
S11	341	-24.4	24.1	-26.5	25.8
S12	337	-24.8	23.9	-25.5	25.1

Table B.13: Tuple Range 21-30 % d_{1a} and % d_{2a} Values for NSERC Tasks

Strategy	n	% d_{1a}	s_1	% d_{2a}	s_2
S1	106	-49.9	28.5	-48.3	28.6
S2	129	-53.6	24.9	-57.4	23.7
S3	168	-37.0	21.5	-39.4	21.7
S4	149	-14.4	23.4	-15.0	23.0
S5	156	-20.5	27.2	-20.2	25.4
S8	244	-81.8	16.3	-80.4	16.9
S9	164	-24.2	21.4	-25.8	22.5
S10	158	-25.8	21.9	-26.2	22.5
S11	143	-12.1	18.2	-13.6	19.9
S12	137	-12.8	18.5	-12.9	18.9

Table B.14: Tuple Range 31-40 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks

Strategy	n	$\%d_{1a}$	s_1	$\%d_{2a}$	s_2
S1	126	-43.8	26.0	-43.2	26.6
S2	158	-56.2	25.8	-59.8	25.1
S3	141	-30.4	23.4	-31.6	23.9
S4	140	-15.9	27.0	-16.4	26.8
S5	158	-26.0	26.9	-24.2	24.5
S8	175	-78.5	20.2	-76.9	20.9
S9	168	-20.9	19.6	-21.6	20.3
S10	174	-22.6	21.4	-22.0	20.2
S11	152	-13.1	19.0	-14.4	20.7
S12	156	-12.8	18.8	-12.9	18.6

Table B.15: Tuple Range 41-50 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks

Strategy	n	$\%d_{1a}$	s_1	$\%d_{2a}$	s_2
S1	162	-51.6	26.7	-49.2	27.1
S2	150	-52.5	22.4	-56.7	21.8
S3	178	-33.8	25.1	-34.8	25.3
S4	173	-23.3	32.1	-22.9	31.5
S5	147	-33.3	31.1	-31.5	29.4
S8	143	-76.3	17.3	-75.1	17.6
S9	170	-23.8	20.6	-24.0	21.8
S10	184	-24.5	20.6	-23.4	20.6
S11	153	-15.3	18.8	-15.5	20.7
S12	154	-15.4	18.9	-14.7	19.5

Table B.16: Tuple Range 51-60 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks

Strategy	n	$\%d_{1a}$	s_1	$\%d_{2a}$	s_2
S1	101	-49.7	25.7	-46.6	25.8
S2	147	-51.7	20.0	-56.3	19.1
S3	123	-27.1	24.5	-29.3	25.9
S4	127	-28.4	31.2	-27.6	30.6
S5	143	-36.9	30.8	-34.1	28.9
S8	159	-76.9	18.9	-75.4	20.0
S9	117	-16.7	18.0	-16.3	19.3
S10	112	-17.0	18.3	-15.6	18.4
S11	100	-11.2	16.0	-10.8	16.9
S12	93	-10.9	16.6	-9.7	15.4

Table B.17: Tuple Range 61-70 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks

Strategy	n	$\%d_{1a}$	s_1	$\%d_{2a}$	s_2
S1	72	-53.2	24.6	-49.2	25.2
S2	122	-52.4	23.6	-57.2	22.6
S3	76	-22.0	22.4	-23.1	23.6
S4	77	-29.2	31.4	-26.3	29.5
S5	97	-40.4	24.9	-36.5	22.6
S8	105	-75.8	19.2	-75.1	20.0
S9	89	-17.2	19.1	-16.4	18.9
S10	89	-16.3	19.0	-14.6	17.6
S11	106	-7.9	15.4	-8.3	16.8
S12	112	-8.0	15.7	-7.7	15.5

Table B.18: Tuple Range 71-80 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks

Strategy	n	$\%d_{1a}$	s_1	$\%d_{2a}$	s_2
S1	83	-14.7	20.8	-14.7	21.1
S2	84	-13.8	18.8	-15.5	22.0
S3	101	-15.2	23.3	-15.5	23.0
S4	104	-15.3	19.0	-16.4	20.7
S5	110	-20.8	21.3	-23.0	23.7
S8	84	-29.0	30.3	-27.7	28.5
S9	88	-40.4	30.4	-36.3	27.5
S10	62	-56.5	24.4	-52.4	24.9
S11	106	-56.0	23.1	-60.3	21.6
S12	113	-76.6	21.7	-76.1	21.6

Table B.19: Tuple Range 81-90 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks

Strategy	n	$\%d_{1a}$	s_1	$\%d_{2a}$	s_2
S1	65	-55.7	29.5	-52.3	30.0
S2	62	-46.4	20.5	-52.0	20.4
S3	79	-20.0	24.1	-21.6	26.3
S4	63	-32.6	35.9	-30.4	34.5
S5	65	-40.1	30.9	-35.7	28.4
S8	65	-75.4	19.3	-76.1	18.4
S9	67	-15.4	19.4	-16.9	22.1
S10	70	-15.4	19.7	-14.7	19.1
S11	50	-7.7	16.2	-8.8	18.6
S12	54	-7.8	16.0	-7.8	14.9

Table B.20: Tuple Range 91-100 $\%d_{1a}$ and $\%d_{2a}$ Values for NSERC Tasks

Strategy	n	$\%d_{1a}$	s_1	$\%d_{2a}$	s_2
S1	45	-52.0	26.9	-49.2	27.3
S2	46	-51.2	26.3	-55.8	24.3
S3	69	-17.7	21.2	-18.7	21.8
S4	72	-21.4	28.2	-19.3	25.1
S5	54	-29.5	30.1	-26.4	27.0
S8	62	-76.8	20.2	-77.2	19.3
S9	53	-15.8	18.6	-17.2	20.1
S10	57	-16.7	18.8	-17.7	19.6
S11	53	-10.4	16.7	-11.7	19.6
S12	54	-10.4	18.1	-10.7	17.9

A.2 Summaries of the Statistical Comparisons of $\%d_{1a}$ and $\%d_{2a}$ Values

To compare the effectiveness of the various strategies, the $\%d_{1a}$ and $\%d_{2a}$ values were ranked in descending order for each tuple range examined. All $\%d_{1a}$ and $\%d_{2a}$ values are negative, so in comparing the effectiveness of two strategies, the strategy with the less negative value is ranked higher. The $\%d_{1a}$ (or $\%d_{2a}$) values for each strategy in a given tuple range were then compared to the $\%d_{1a}$ (or $\%d_{2a}$) values for all strategies that had a lower ranking in same tuple range.

By the central limit theorem, the means of m identically distributed independent random variables, each with a mean μ and finite variance σ^2 , will be approximately normally distributed, regardless of the underlying distribution of the individual variables. Since the sample size is greater than 30 in all cases, the z-test was used to determine the significance of the difference in mean values.

Consider the population of $\%d_{1a}$ values for strategy j with a mean value μ_{1j} . Let the $\%d_{1a}$, s_1 and n values for strategy j be written $\%d_{1aj}$, s_{1j} and n_j . Let the μ_{1k} , $\%d_{1ak}$, s_{1k} and n_k values for strategy k be written μ_{1k} , $\%d_{1ak}$, s_{1k} and n_k . The equation for the z-test

can be written in the form

$$z = (| \%d_{1aj} - \%d_{1ak} |) / (s_{1j}^2/n_j + s_{1k}^2/n_k)^{1/2} \quad \dots (B.5)$$

The null hypothesis is that μ_{1j} is equal to μ_{1k} . The alternative hypothesis is that μ_{1j} is less negative than μ_{1k} . Similarly, when the $\%d_{2a}$ values for the two strategies are compared, the equation for the z-test can be written in the form

$$z = (| \%d_{2aj} - \%d_{2ak} |) / (s_{2j}^2/n_j + s_{2k}^2/n_k)^{1/2} \quad \dots (B.6)$$

The null hypothesis is that μ_{2j} is equal to μ_{2k} . The alternative hypothesis is that μ_{2j} is less negative than μ_{2k} . A one tailed test with a significance level of 5% is used. The null hypothesis is accepted if $z < 1.645$.

To provide an example of the statistical calculation, the $\%d_{1a}$ values for strategy S9 and strategy S10 for SaskTel task relations with a tuple range 1-10 are compared. For strategy S9, $\%d_{1a} = -57.4894\%$, s_1 is 21.9195 and $n = 1038$. For strategy S10, $\%d_{1a} = -61.8419\%$, $s_1 = 23.2985$ and $n = 860$. Since $\%d_{1a}$ for strategy S9 is less negative than $\%d_{1a}$ for strategy S10, strategy S9 is ranked higher than strategy S10. Using equation B.5, the value of z is determined as

$$z = |-57.4894 - (-61.8419)| / ((21.9195)^2/1038 + (23.2985)^2/860)^{1/2} = 4.161$$

Since $z > 1.645$, the null hypothesis is rejected and the alternative hypothesis is accepted. Thus, μ_l for strategy S9 is significantly less negative than μ_l for strategy S10 at the 5% level of significance. For conciseness, statistical comparisons in this thesis will be described in terms of $\%d_{1a}$ and $\%d_{2a}$ values, rather than population means.

In the following statistical summary tables, the leftmost column and the topmost row contain the strategy names in ranked order. The values in the table are z values obtained by comparing the $\%d_{1a}$ (or $\%d_{2a}$) values for each strategy to the $\%d_{1a}$ (or $\%d_{2a}$) values of all lower ranked strategies in the same tuple range. A blank element in the tables indicates that the strategy labeling the row was ranked higher than the strategy labeling the column or that the strategy labeling the row is the same as the strategy labeling the column. If the null hypothesis is accepted, the z value in the table is shown in bold print. Tables B.21 to B.30 contain information about the comparison of $\%d_{1a}$ values for the SaskTel tasks. Tables B.31 to B.40 have similar information about the NSERC tasks. Tables B.41 to B.50 contain information about the comparison of $\%d_{2a}$ values for the Sasktel tasks and Tables B.51 to B.60 shows the comparison of $\%d_{2a}$ for the NSERC tasks.

Table B.21: Tuple Range 1-10 z Values for $\%d_{1a}$ Comparison for SaskTel Tasks

	S4	S12	S11	S1	S5	S2	S9	S10	S3
S4									
S12	12.7789								
S11	12.8915	0.1988							
S1	19.0482	3.7369	2.5849						
S5	27.3028	9.5403	6.3649	4.2429					
S2	33.8100	22.5121	18.5348	18.0013	15.9389				
S9	62.2052	50.5779	41.4409	43.2861	42.8943	21.0289			
S10	59.6814	49.4761	42.1507	43.4996	42.8828	23.4950	4.1612		
S3	68.9829	57.9523	48.3010	50.5561	50.5009	28.1846	7.9422	3.2950	
S8	91.8290	78.0652	60.2234	65.4407	67.5200	35.9942	13.2117	7.4524	3.7823

Table B.22: Tuple Range 11-20 z Values for $\%d_{1a}$ Comparison SaskTel Tasks

	S12	S11	S4	S1	S5	S9	S10	S3	S2
S12									
S11	0.8773								
S4	11.7029	10.6841							
S1	22.4168	21.4773	12.9676						
S5	27.2730	26.1325	16.8033	1.1960					
S9	21.7143	20.8679	13.0248	1.1994	0.2604				
S10	22.9909	22.1630	14.5806	2.9237	2.1953	1.6371			
S3	32.1043	31.3057	24.6370	13.0081	13.2577	11.2123	9.4958		
S2	42.0259	40.9093	33.4852	17.1412	18.6011	14.3842	12.1870	0.7014	
S8	63.8099	62.6843	57.1501	39.6231	43.8163	35.3577	32.8173	20.6502	24.9009

Table B.23: Tuple Range 21-30 z Values for $\%d_{1a}$ Comparison SaskTel Tasks

	S11	S12	S4	S1	S9	S5	S10	S2	S3
S11									
S12	0.7946								
S4	12.7151	11.3102							
S1	18.7093	17.2070	6.8183						
S9	18.5831	17.1448	7.0197	0.3511					
S5	24.9835	22.6892	10.5760	2.3604	1.8938				
S10	22.0864	20.4503	10.1195	3.1036	2.6850	1.1935			
S2	34.7825	32.0821	20.4420	11.6574	10.9526	11.1006	8.1646		
S3	28.7534	27.0079	17.2790	10.2053	9.6673	9.2811	7.2309	0.2760	
S8	52.0874	48.4889	37.0282	26.9227	25.8146	29.1310	23.3848	17.4522	17.1192

Table B.24: Tuple Range 31-40 z Values for $\%d_{1a}$ Comparison SaskTel Tasks

	S11	S12	S4	S9	S1	S10	S5	S3	S2
S11									
S12	0.5904								
S4	17.8980	17.2095							
S9	26.9754	25.7924	3.7790						
S1	26.8982	25.7658	4.1619	0.4827					
S10	29.2025	28.0799	6.4855	3.1584	2.6487				
S5	34.9145	33.5322	9.6079	6.5853	6.0061	3.1975			
S3	27.9666	27.2575	10.0705	7.5044	7.0423	4.7558	2.1891		
S2	38.7914	37.4302	13.4883	10.9837	10.3639	7.5499	4.5811	1.6046	
S8	65.8873	63.5132	30.5515	30.1564	29.1961	25.7705	23.2360	16.4825	17.7948

Table B.25: Tuple Range 41-50 z Values for $\%d_{Ia}$ Comparison SaskTel Tasks

	S11	S12	S9	S10	S4	S1	S3	S5	S2
S11									
S12	1.2493								
S9	19.1202	17.4711							
S10	19.1745	17.7677	1.8140						
S4	16.8043	15.8964	3.6680	2.1136					
S1	23.8656	22.5943	7.9518	5.9751	3.0711				
S3	33.5918	31.3516	11.5832	8.8849	4.9200	1.6702			
S5	32.7781	31.0561	14.0112	11.4949	7.4633	4.6919	3.6749		
S2	35.4030	33.6046	16.2410	13.5873	9.2002	6.6094	5.9106	2.0389	
S8	84.8018	77.7104	41.7045	35.3330	24.4263	23.7751	28.0255	19.8358	17.2772

Table B.26: Tuple Range 51-60 z Values for $\%d_{Ia}$ Comparison SaskTel Tasks

	S11	S12	S9	S10	S4	S1	S3	S5	S2
S11									
S12	1.2202								
S9	4.4836	2.8783							
S10	6.4835	4.8364	2.3157						
S4	12.9657	11.2631	9.5398	7.2900					
S1	16.5160	14.4742	12.9452	10.3931	2.3637				
S3	21.3494	18.6360	17.4688	14.3141	4.9394	2.5187			
S5	21.7755	19.3193	18.1796	15.2832	6.4047	4.1837	1.9823		
S2	26.9481	23.5128	22.9324	19.1571	8.4539	6.1365	3.9104	1.5117	
S8	40.5887	35.5632	36.6549	31.5505	18.0238	16.2545	15.3011	11.7597	12.0790

Table B.27: Tuple Range 61-70 z Values for $\%d_{Ia}$ Comparison SaskTel Tasks

	S11	S12	S9	S10	S3	S4	S1	S5	S2
S11									
S12	1.7381								
S9	8.2745	5.9107							
S10	10.2953	7.9991	2.4523						
S3	14.8526	12.9374	8.4078	6.1663					
S4	15.3332	13.5948	9.3889	7.2789	1.3344				
S1	18.7630	16.6749	12.1319	9.7746	3.2197	1.7398			
S5	26.8614	23.7543	18.3863	15.3820	7.3061	5.4723	3.7763		
S2	29.8049	26.2256	20.5174	17.2419	8.5693	6.6052	4.9276	1.1581	
S8	41.3390	37.1349	31.3238	27.6265	17.6136	15.1786	13.9416	11.1284	10.3243

Table B.28: Tuple Range 71-80 z Values for $\%d_{1a}$ Comparison SaskTel Tasks

	S11	S12	S9	S10	S4	S1	S3	S2	S5
S11									
S12	1.3645								
S9	13.9132	12.3393							
S10	19.7019	17.9002	4.7543						
S4	11.2543	10.4557	3.0625	0.1324					
S1	14.2425	13.4029	5.8248	2.8866	2.1488				
S3	26.0571	24.1140	10.6262	6.0633	3.6025	0.9029			
S2	22.5471	21.3030	11.3678	7.8093	5.4312	3.0356	3.0839		
S5	23.5856	22.3741	12.6836	9.2342	6.5356	4.1823	4.6252	1.3611	
S8	51.4344	48.2830	30.7902	26.3112	15.2656	12.6067	19.3599	11.0573	9.3617

Table B.29: Tuple Range 81-90 z Values for $\%d_{1a}$ Comparison SaskTel Tasks

	S11	S12	S9	S10	S4	S3	S1	S2	S5
S11									
S12	1.8601								
S9	7.3712	5.1305							
S10	8.5164	6.3906	1.5178						
S4	11.1833	9.9120	6.6641	5.5087					
S3	18.0445	16.0047	11.7427	10.1356	2.6860				
S1	19.8302	17.9881	14.0798	12.5635	5.0941	2.8743			
S2	22.3549	20.2537	16.0477	14.3811	6.2109	4.0953	1.0603		
S5	25.0771	22.7168	18.2405	16.4265	7.5391	5.5870	2.3921	1.3521	
S8	34.1631	31.1581	26.1217	23.9307	12.9161	11.7647	8.1008	7.2315	5.9906

Table B.30: Tuple Range 91-100 z Values for $\%d_{1a}$ Comparison SaskTel Tasks

	S11	S9	S12	S10	S3	S4	S2	S1	S5
S11									
S9	1.3375								
S12	2.6358	1.4996							
S10	5.0747	3.9182	1.9711						
S3	11.9505	11.1366	8.6898	7.5495					
S4	14.6752	14.0390	12.1302	11.3150	5.2800				
S2	19.0222	18.4637	16.0973	15.5272	8.8649	2.7555			
S1	21.0467	20.5336	17.7360	17.3261	9.9896	3.2919	0.4216		
S5	17.6556	17.0937	15.1631	14.5124	8.6692	3.1633	0.6382	0.2826	
S8	36.8785	37.2517	30.6812	32.4822	20.6555	9.5574	6.5867	6.6249	5.0213

Table B.31: Tuple Range 1-10 z Values for $\%d_{la}$ Comparison for NSERC Tasks

	s4	s5	s1	s11	s12	s2	s9	s10	s8
s4									
s5	0.9852								
s1	8.5413	7.5816							
s11	11.7718	10.8607	3.4440						
s12	11.7905	10.8795	3.4626	0.0182					
s2	12.3293	11.4101	3.8959	0.4070	0.3887				
s9	17.7496	16.8296	8.9781	5.2748	5.2559	4.9033			
s10	18.7961	17.8688	9.8912	6.1233	6.1042	5.7561	0.8218		
s3	26.6540	25.6106	16.0398	11.5438	11.5221	11.1939	5.7752	4.9103	
s8	34.2559	33.1856	22.7523	17.7940	17.7710	17.4978	11.9770	11.1465	6.1363

Table B.32: Tuple Range 11-20 z Values for $\%d_{la}$ Comparison NSERC Tasks

	s4	s5	s11	s12	s9	s10	s3	s1	s2
s4									
s5	1.1628								
s11	6.5316	5.0827							
s12	6.7306	5.2755	0.2186						
s9	12.3629	10.7893	6.5949	6.3845					
s10	13.0724	11.5156	7.4835	7.2782	1.0175				
s3	16.7066	15.0719	11.5141	11.3123	5.0176	3.9266			
s1	16.1284	14.7178	11.3884	11.2118	5.7161	4.7530	1.2774		
s2	17.9942	16.5244	13.3468	13.1683	7.5605	6.5535	3.0209	1.5627	
s8	39.2509	36.3566	36.5087	36.2651	27.3288	25.2362	20.0881	15.3809	13.6144

Table B.33: Tuple Range 21-30 z Values for $\%d_{la}$ Comparison NSERC Tasks

	s11	s12	s4	s5	s9	s10	s3	s1	s2
s11									
s12	0.3371								
s4	0.9651	0.6519							
s5	3.1813	2.8634	2.0999						
s9	5.3814	4.9604	3.8566	1.3539					
s10	5.9498	5.5301	4.4037	1.9054	0.6608				
s3	11.0786	10.5487	8.9089	6.0209	5.4165	4.6398			
s1	11.9768	11.6292	10.5357	8.3388	7.9300	7.3522	3.9908		
s2	15.5623	15.0792	13.4502	10.6969	10.6370	9.9023	6.0231	1.0409	
s8	37.8015	36.3716	30.8817	25.3660	29.1715	27.5255	22.8359	10.7854	9.5435

Table B.34: Tuple Range 31-40 z Values for $\%d_{la}$ Comparison NSERC Tasks

	s12	s11	s4	s9	s10	s5	s3	s1	s2
s12									
s11	0.1167								
s4	1.1177	1.0185							
s9	3.7877	3.6280	1.8366						
s10	4.4455	4.2858	2.4219	0.7904					
s5	5.0362	4.9026	3.2376	1.9483	1.2500				
s3	7.0889	6.9281	4.8159	3.8233	3.0351	1.5048			
s1	11.2095	11.0426	8.5888	8.2804	7.4841	5.6418	4.4179		
s2	17.0402	16.8055	13.1438	13.8556	12.8388	10.1843	9.0887	4.0080	
s8	30.5783	30.1191	22.7914	26.7832	25.0672	19.9406	19.3074	12.4852	8.6943

Table B.35: Tuple Range 41-50 z Values for $\%d_{la}$ Comparison NSERC Tasks

	s11	s12	s4	s9	s10	s5	s3	s1	s2
s11									
s12	0.0656								
s4	2.7796	2.7315							
s9	3.8953	3.8333	0.1896						
s10	4.2832	4.2202	0.4217	0.3015					
s5	6.0468	6.0015	2.8358	3.1495	2.9618				
s3	7.6670	7.6125	3.4254	4.0718	3.8651	0.1615			
s1	14.0026	13.9542	8.7934	10.5637	10.4594	5.5106	6.2984		
s2	15.6642	15.6129	9.5968	11.8815	11.8047	6.1027	7.1330	0.3446	
s8	29.0746	29.0266	18.6864	24.4976	24.7149	14.5897	17.8962	9.6888	10.1833

Table B.36: Tuple Range 51-60 z Values for $\%d_{la}$ Comparison NSERC Tasks

	s12	s11	s9	s10	s3	s4	s5	s1	s2
s12									
s11	0.1251								
s9	2.3910	2.3541							
s10	2.4726	2.4377	0.1277						
s3	5.7578	5.8069	3.7562	3.5944					
s4	5.3658	5.3806	3.6391	3.5076	0.3844				
s5	8.3866	8.4750	6.6009	6.4267	2.9051	2.2475			
s1	12.5501	12.7324	10.7988	10.5754	6.6836	5.6341	3.5093		
s2	17.1134	17.6411	14.9536	14.5359	8.9407	7.2257	4.8309	0.6620	
s8	28.8913	29.9657	26.8664	26.1682	18.6780	15.4135	13.4271	9.1913	11.3513

Table B.37: Tuple Range 61-70 z Values for $\%d_{1a}$ Comparison NSERC Tasks

	s11	s12	s10	s9	s3	s4	s5	s2	s1
s11									
s12	0.0703								
s10	3.3613	3.3140							
s9	3.6959	3.6498	0.2989						
s3	4.7312	4.6931	1.7218	1.4592					
s4	5.4855	5.4553	3.1267	2.9176	1.6405				
s5	11.0599	11.0378	7.4363	7.1663	5.1112	2.5507			
s2	17.0538	17.0526	12.2713	11.9688	9.1073	5.5598	3.6307		
s1	13.8908	13.8742	10.4402	10.1912	8.0648	5.2063	3.3318	0.2204	
s8	28.2522	28.2993	21.5681	21.2343	16.9127	11.5159	11.2489	8.2285	6.5503

Table B.38: Tuple Range 71-80 z Values for $\%d_{1a}$ Comparison NSERC Tasks

	s11	s12	s10	s9	s3	s4	s5	s2	s1
s11									
s12	0.2743								
s10	0.4215	0.1602							
s9	0.5226	0.2056	0.0160						
s3	2.4245	2.0123	1.6938	2.0089					
s4	3.9140	3.5781	3.2837	3.6265	2.1186				
s5	6.9445	6.5011	6.0643	6.7287	5.1264	2.4584			
s2	13.8882	12.9109	11.8597	13.9498	11.6137	6.7445	3.9538		
s1	11.4852	10.8606	10.1885	11.3874	9.6165	6.0490	3.5770	0.1024	
s8	21.7398	20.2357	18.5542	22.1883	19.3685	12.2351	9.4470	5.5462	5.4251

Table B.39: Tuple Range 81-90 z Values for $\%d_{1a}$ Comparison NSERC Tasks

	s11	s12	s9	s10	s3	s4	s5	s2	s1
s11									
s12	0.0547								
s9	2.3516	2.3550							
s10	2.3629	2.3668	0.0030						
s3	3.4662	3.4888	1.2602	1.2611					
s4	4.9088	4.9237	3.3543	3.3573	2.3899				
s5	7.2761	7.3320	5.4844	5.4925	4.2999	1.2736			
s2	11.1548	11.3478	8.7862	8.8103	7.0348	2.6498	1.3596		
s1	11.1253	11.2404	9.2351	9.2513	7.8507	3.9721	2.9413	2.0642	
s8	20.4106	20.8485	17.7748	17.8314	15.3253	8.3571	7.8105	8.1791	4.5048

Table B.40: Tuple Range 91-100 z Values for $\%d_{1a}$ Comparison NSERC Tasks

	s11	s12	s9	s10	s3	s4	s5	s2	s1
s11									
s12	0.0033								
s9	1.5588	1.5052							
s10	1.8558	1.7908	0.2600						
s3	2.1251	2.0527	0.5332	0.2796					
s4	2.7196	2.6521	1.3397	1.1290	0.8821				
s5	4.0668	3.9920	2.8409	2.6676	2.4451	1.5350			
s2	9.0426	8.8662	7.6134	7.4717	7.2100	5.8290	3.8474		
s1	9.0211	8.8531	7.6307	7.4911	7.2361	5.8917	3.9440	0.1600	
s8	19.2782	18.6523	16.8220	16.7812	16.3315	13.1910	9.7967	5.5059	5.1955

Table B.41: Tuple Range 1-10 z Values for $\%d_{2a}$ Comparison for SaskTel Tasks

	S4	S1	S5	S12	S11	S2	S9	S10	S3
S4									
S1	13.8187								
S5	20.8585	3.5228							
S12	21.5199	11.6294	9.9448						
S11	23.4819	13.9008	12.3816	2.2509					
S2	26.1365	16.7973	15.4528	5.0451	2.7840				
S9	65.4556	53.2054	53.7003	35.0695	32.0371	28.5119			
S10	56.5592	47.1067	46.9559	32.7294	30.1372	27.0678	1.3981		
S3	62.5520	52.2843	52.4061	36.5340	33.7696	30.5252	4.1888	2.5076	
S8	80.5096	66.0905	67.5424	44.4084	40.9804	37.0695	7.6552	5.3876	2.7105

Table B.42: Tuple Range 11-20 z Values for $\%d_{2a}$ Comparison SaskTel Tasks

	S12	S11	S4	S1	S5	S10	S9	S2	S8
S12									
S11	2.0622								
S4	2.9293	0.3171							
S1	11.9398	9.1479	11.0576						
S5	14.8475	11.2718	15.0129	0.8514					
S10	19.3481	16.6527	19.2172	9.0419	9.3865				
S9	19.3452	16.6544	19.2110	9.0526	9.3963	0.0198			
S2	31.4565	27.6787	33.4567	19.8636	22.0278	8.5264	8.4945		
S3	27.0893	24.4359	27.4460	17.8062	18.7936	8.8094	8.7837	1.7310	
S8	60.4776	55.2492	66.3244	48.9563	55.0152	34.2859	34.2252	28.5046	21.8144

Table B.43: Tuple Range 21-30 z Values for $\%d_{2a}$ Comparison SaskTel Tasks

	S12	S11	S4	S1	S5	S9	S10	S2	S3
S12									
S11	0.1635								
S4	4.8889	4.4832							
S1	9.9976	9.4654	6.0651						
S5	13.9167	13.0273	9.7174	2.0402					
S9	15.5032	14.8414	12.1424	5.9133	4.8326				
S10	16.9974	16.2457	13.6426	7.0376	6.1074	0.8916			
S2	25.3918	24.2246	22.4990	14.4409	14.7587	7.5015	6.7742		
S3	22.6911	21.9281	19.9410	13.7372	13.5636	7.9533	7.3172	1.6447	
S8	45.6526	43.6735	44.2753	33.7757	37.3745	25.6246	25.4784	19.8384	14.8589

Table B.44: Tuple Range 31-40 z Values for $\%d_{2a}$ Comparison SaskTel Tasks

	S11	S12	S4	S1	S9	S5	S10	S2	S3
S11									
S12	0.0865								
S4	11.1923	11.1742							
S1	16.5869	16.5931	3.3787						
S9	22.7216	22.7464	8.8022	5.8532					
S5	24.1800	24.2170	9.2610	6.2495	0.1730				
S10	23.1844	23.2060	9.7561	6.9401	1.2379	1.1104			
S2	28.6835	28.7188	14.6513	12.1452	6.4229	6.4567	5.0700		
S3	22.6619	22.6633	12.4201	10.1539	5.4426	5.4229	4.3401	0.0473	
S8	58.3540	58.5018	37.3864	36.0748	29.2994	30.1081	27.2011	21.4280	18.0106

Table B.45: Tuple Range 41-50 z Values for $\%d_{2a}$ Comparison SaskTel Tasks

	S11	S12	S4	S9	S10	S1	S3	S5	S2
S11									
S12	0.2547								
S4	10.5223	10.5687							
S9	15.6137	15.9884	0.3388						
S10	15.4595	15.7810	0.7478	0.5459					
S1	16.3044	16.5043	3.3930	3.8153	3.2644				
S3	25.7260	26.4074	7.3725	9.2692	8.4072	3.9218			
S5	23.2507	23.6615	7.6908	9.2745	8.5350	4.4972	1.0194		
S2	27.2763	27.7362	11.1664	13.4806	12.6619	8.2905	5.4209	4.0671	
S8	68.6935	71.9790	29.8247	41.3304	38.8175	28.4785	29.6264	24.5748	18.7246

Table B.46: Tuple Range 51-60 z Values for $\%d_{2a}$ Comparison SaskTel Tasks

	S12	S11	S10	S9	S4	S1	S5	S3	S2
S12									
S11	0.6339								
S10	3.6697	2.8043							
S9	3.7616	2.9280	0.2424						
S4	7.5831	6.7370	4.7288	4.3983					
S1	10.7385	9.6969	7.8535	7.3981	2.4043				
S5	15.2050	13.9437	12.4618	11.8457	6.2047	3.9580			
S3	16.7257	15.3424	13.9838	13.2853	7.2887	5.0517	1.0133		
S2	19.2682	17.6307	16.5211	15.6489	8.9344	6.6986	2.4819	1.4572	
S8	35.3178	32.5075	33.3629	31.5813	21.6054	20.0807	15.8286	15.1798	14.5989

Table B.47: Tuple Range 61-70 z Values for $\%d_{2a}$ Comparison SaskTel Tasks

	S11	S12	S9	S10	S4	S3	S1	S5	S2
S11									
S12	0.1899								
S9	7.0961	7.1742							
S10	7.8956	8.0313	0.2143						
S4	10.3062	10.3576	5.0657	5.0637					
S3	11.7487	11.8474	5.9179	5.9591	0.3635				
S1	13.1976	13.3210	7.2822	7.3699	1.4839	1.1866			
S5	18.4149	18.6819	11.5431	11.8282	4.5091	4.3945	3.1392		
S2	20.7942	21.1309	13.5300	13.9142	5.9396	5.9211	4.6434	1.5431	
S8	37.8420	38.5452	29.1007	30.1192	18.2143	18.9915	17.6788	15.3210	14.0032

Table B.48: Tuple Range 71-80 z Values for $\%d_{2a}$ Comparison SaskTel Tasks

	S12	S11	S9	S4	S10	S1	S3	S2	S5
S12									
S11	0.1170								
S9	12.4880	11.2445							
S4	7.4142	7.1296	0.5151						
S10	16.6787	14.9128	3.0044	1.2734					
S1	10.4664	10.0860	3.3850	2.2462	1.6281				
S3	22.6689	20.5809	9.0131	5.0575	6.3760	2.2245			
S2	17.1735	16.3548	8.2368	5.5764	6.2575	3.1186	1.6080		
S5	17.6029	16.8049	8.8720	6.1159	6.9473	3.6900	2.3887	0.6826	
S8	47.9225	43.2234	29.7604	17.4485	27.7992	14.6153	20.0263	12.9319	11.7969

Table B.49: Tuple Range 81-90 z Values for $\%d_{2a}$ Comparison SaskTel Tasks

	S11	S12	S10	S9	S4	S3	S1	S2	S5
S11									
S12	0.4664								
S10	6.2327	5.8032							
S9	6.2759	5.8487	0.0669						
S4	7.3041	7.0668	3.5531	3.5053					
S3	15.6674	15.3412	10.4934	10.4193	4.5895				
S1	15.0515	14.7770	10.6442	10.5809	5.3424	1.1885			
S2	16.5618	16.2888	12.1584	12.0938	6.6042	2.6363	1.3579		
S5	18.8456	18.5386	13.9003	13.8260	7.5308	3.5035	2.0894	0.6624	
S8	32.0617	31.6817	25.8196	25.7130	15.9850	12.9584	10.6926	9.1142	8.8703

Table B.50: Tuple Range 91-100 z Values for $\%d_{2a}$ Comparison SaskTel Tasks

	S11	S12	S9	S10	S3	S4	S1	S5	S2
S11									
S12	0.9424								
S9	1.0149	0.0755							
S10	2.4530	1.5541	1.4757						
S3	9.4121	9.0339	8.9812	8.3000					
S4	11.0201	10.6933	10.6509	10.0904	3.3945				
S1	15.3215	15.2027	15.1631	14.8397	6.7508	2.4979			
S5	13.2617	13.0063	12.9678	12.5036	5.9500	2.3615	0.1338		
S2	15.3348	15.1734	15.1355	14.7729	7.4587	3.4180	1.1509	0.8997	
S8	33.6263	35.2534	35.2613	37.4013	21.9406	13.1062	11.5744	9.4766	9.0910

Table B.51: Tuple Range 1-10 z Values for $\%d_{2a}$ Comparison for NSERC Tasks

	s4	s5	s1	s12	s11	s2	s9	s10	s3
s4									
s5	0.7613								
s1	10.6010	9.5698							
s12	11.3504	10.3549	1.1103						
s11	12.1396	11.1283	1.8871	0.7565					
s2	13.4919	12.4011	2.8524	1.6636	0.8871				
s9	18.2450	16.9940	7.0953	5.7535	4.9534	4.1490			
s10	18.6160	17.3375	7.3366	5.9758	5.1691	4.3647	0.1886		
s3	27.7442	25.8168	13.8175	12.0099	11.0812	10.3574	5.7676	5.6053	
s8	34.0227	31.7548	18.9753	16.9009	15.9247	15.3217	10.6763	10.5538	5.5307

Table B.52: Tuple Range 11-20 z Values for $\%d_{2a}$ Comparison NSERC Tasks

	s4	s5	s12	s11	s9	s10	s1	s3	s2
s4									
s5	0.9092								
s12	6.6049	5.6293							
s11	7.0320	6.0661	0.5281						
s9	12.6549	11.6959	6.6092	6.0250					
s10	12.8792	11.9394	6.9651	6.3912	0.4857				
s1	15.9166	15.0618	10.6809	10.1396	4.7537	4.2451			
s3	17.8006	16.8415	12.1065	11.4714	5.4100	4.8221	0.0965		
s2	18.7782	17.9057	13.6006	13.0241	7.5099	6.9478	2.4339	2.5645	
s8	39.2850	38.0096	34.1418	32.9173	24.5728	23.2397	15.1291	17.4960	12.2614

Table B.53: Tuple Range 21-30 z Values for $\%d_{2a}$ Comparison NSERC Tasks

	s12	s11	s4	s5	s9	s10	s3	s1	s2
s12									
s11	0.2878								
s4	0.8471	0.5698							
s5	2.7922	2.5022	1.8561						
s9	5.3822	5.0251	4.1689	2.0847					
s10	5.5050	5.1498	4.2951	2.2215	0.1632				
s3	11.3646	10.9010	9.6513	7.2865	5.5912	5.3768			
s1	11.0060	10.7066	9.9064	8.1651	6.8407	6.6847	2.7407		
s2	16.8336	16.3743	15.0484	12.7610	11.5623	11.3357	6.7174	2.6243	
s8	34.7463	33.6224	30.0957	26.1626	26.4361	25.9315	20.5504	10.7889	9.7763

Table B.54: Tuple Range 31-40 z Values for $\%d_{2a}$ Comparison NSERC Tasks

	s12	s11	s4	s9	s10	s5	s3	s1	s2
s12									
s11	0.6694								
s4	1.2863	0.7041							
s9	4.0126	3.1274	1.8882						
s10	4.2438	3.3312	2.0418	0.1717					
s5	4.6149	3.8207	2.6266	1.0585	0.9177				
s3	7.4540	6.5532	5.0146	3.9146	3.8044	2.6176			
s1	10.8202	9.9178	8.1834	7.6069	7.5338	6.1789	3.7415		
s2	18.8447	17.4302	14.3989	15.0730	15.0737	12.7536	9.9682	5.3578	
s8	29.4163	27.0860	21.9012	24.8160	24.9673	20.9500	17.6926	11.8076	6.7075

Table B.55: Tuple Range 41-50 z Values for $\%d_{2a}$ Comparison NSERC Tasks

	s12	s11	s4	s10	s9	s5	s3	s1	s2
s12									
s11	0.3550								
s4	2.8802	2.5472							
s10	3.9870	3.5005	0.1661						
s9	4.0596	3.5985	0.3679	0.2670					
s5	5.8182	5.4346	2.5158	2.8303	2.5443				
s3	8.1879	7.6588	3.8982	4.7069	4.2829	1.0784			
s1	13.0422	12.4559	8.2015	9.8633	9.3043	5.4826	5.0495		
s2	17.6581	16.8485	11.2954	14.1935	13.3496	8.3468	8.3933	2.6735	
s8	28.0166	26.7405	18.5419	24.4010	22.8920	15.3372	16.7762	9.9785	7.9699

Table B.56: Tuple Range 51-60 z Values for $\%d_{2a}$ Comparison NSERC Tasks

	s12	s11	s10	s9	s4	s3	s5	s1	s2
s12									
s11	0.4653								
s10	2.5100	1.9991							
s9	2.7746	2.2639	0.2846						
s4	5.6871	5.2651	3.7179	3.4744					
s3	6.9037	6.4036	4.6749	4.3950	0.4570				
s5	8.4366	7.9231	6.2168	5.9311	1.7934	1.4513			
s1	12.2057	11.6583	9.9881	9.6876	5.0778	4.9919	3.5329		
s2	20.8025	19.7385	17.3515	16.8310	9.1373	9.5921	7.6841	3.2212	
s8	29.2295	27.9340	25.4274	24.8075	15.2144	16.3503	14.2985	9.5715	8.5984

Table B.57: Tuple Range 61-70 z Values for $\%d_{2a}$ Comparison NSERC Tasks

	s12	s11	s10	s9	s3	s4	s5	s1	s2
s12									
s11	0.3060								
s10	2.9006	2.5047							
s9	3.5145	3.1157	0.6739						
s3	4.9914	4.6487	2.5853	1.9766					
s4	5.0815	4.8071	3.0599	2.5349	0.7547				
s5	10.6008	10.0103	7.4349	6.6114	3.7924	2.5074			
s1	12.5411	12.0567	9.8813	9.1577	6.4989	5.0972	3.3741		
s2	19.6777	18.6654	15.4130	14.2588	10.0624	7.8558	6.7400	2.2404	
s8	27.6061	26.2169	22.4180	20.9784	15.5694	12.5386	12.7981	7.2913	6.2931

Table B.58: Tuple Range 71-80 z Values for $\%d_{2a}$ Comparison NSERC Tasks

	s12	s11	s10	s9	s3	s4	s5	s1	s2
s12									
s11	0.2442								
s10	0.2464	0.0110							
s9	0.5779	0.3074	0.2830						
s3	2.5641	2.2704	2.1739	2.1430					
s4	3.3598	3.1098	3.0168	3.0265	1.2316				
s5	5.7972	5.5029	5.3349	5.5704	3.6108	2.0196			
s1	9.6289	9.3039	9.0502	9.5631	7.5765	5.5726	3.7293		
s2	14.5967	14.0570	13.4836	14.9985	12.0969	8.6768	6.6309	2.0578	
s8	19.9568	19.2979	18.4945	20.7648	17.4938	13.0279	11.1419	6.2883	5.4256

Table B.59: Tuple Range 81-90 z Values for $\%d_{2a}$ Comparison NSERC Tasks

	s12	s11	s10	s9	s3	s4	s5	s2	s1
s12									
s11	0.3073								
s10	2.2635	1.6864							
s9	2.7164	2.1570	0.6383						
s3	3.8720	3.2440	1.8676	1.1801					
s4	4.7203	4.2515	3.2064	2.6375	1.6683				
s5	6.8925	6.1299	5.0251	4.2454	3.0656	0.9489			
s2	13.4584	11.6889	10.8131	9.3791	7.7140	4.2536	3.7163		
s1	10.5320	9.5555	8.6366	7.7119	6.4587	3.8260	3.2414	0.0786	
s8	22.3823	19.2864	19.0181	16.7398	14.5601	9.2874	9.6179	6.9861	5.4481

Table B.60: Tuple Range 91-100 z Values for $\%d_{2a}$ Comparison NSERC Tasks

	s12	s11	s9	s10	s3	s4	s5	s1	s2
s12									
s11	0.2898								
s9	1.7759	1.4228							
s10	1.9653	1.5882	0.1162						
s3	2.2476	1.8599	0.3929	0.2870					
s4	2.2603	1.9015	0.5205	0.4242	0.1541				
s5	3.5721	3.2249	1.9979	1.9463	1.7034	1.5018			
s1	8.1271	7.6825	6.5003	6.5401	6.2972	5.9423	4.1631		
s2	10.4206	9.8372	8.5244	8.6289	8.3526	7.8570	5.7372	1.2210	
s8	19.2575	17.9809	16.2183	16.6883	16.2803	15.0692	11.5077	5.8853	4.9116

IMAGE EVALUATION TEST TARGET (QA-3)

APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved