2001

The Extraction of Classification Rules and Decision Trees from Independence Diagrams

Robert W. Kerbs
Nova Southeastern University, rwk@robertkerbs.com

The Extraction of Classification Rules and Decision Trees from Independence Diagrams

by

Robert W. Kerbs

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

The Graduate School of Computer and Information Sciences
Nova Southeastern University

2001

We hereby certify that this dissertation, submitted by Robert W. Kerbs, conforms to acceptable standards and is fully adequate in scope and quality to fulfill the dissertation requirements for the degree of Doctor of Philosophy.

_____ _____

Junping Sun, Ph.D. Date
Chairperson of Dissertation Committee

_____ _____

Michael J. Laszlo, Ph.D. Date
Dissertation Committee Member

_____ _____

Lee J. Leitner, Ph.D. Date
Dissertation Committee Member

Approved:

_____ _____

Edward Lieblein, Ph.D. Date
Dean, The Graduate School of Computer and Information Sciences

The Graduate School of Computer and Information Sciences
Nova Southeastern University

2001

An Abstract of a Dissertation Submitted to Nova Southeastern University
in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

The Extraction of Classification Rules and Decision Trees from Independence Diagrams

by

Robert W. Kerbs

2001

Databases are growing exponentially in many application domains. Timely construction of models that represent identified patterns and regularities in the data facilitate the prediction of future events based upon past performance. Data mining can promote this process through various model building techniques. The goal is to create models that intuitively represent the data and perhaps aid in the discovery of new knowledge.

Most data mining methods rely upon either fully-automated information-theoretic or statistical algorithms. Typically, these algorithms are non-interactive, hide the model derivation process from the user, require the assistance of a domain expert, are application-specific, and may not clearly translate detected relationships.

This paper proposes a visual data mining algorithm, BLUE, as an alternative to present data mining techniques. BLUE visually supports the processes of classification and prediction by combining two visualization methods. The first consists of a modification to independence diagrams, called BIDS, allowing for the examination of pairs of categorical attributes in relational databases. The second uses decision trees to provide a global context from which a model can be constructed. Classification rules are extracted from the decision trees to assist in concept representation.

BLUE uses the abilities of the human visual system to detect patterns and regularities in images. The algorithm employs a mechanism that permits the user to interactively backtrack to previously visited nodes to guide and explore the creation of the model. As a decision tree is induced, classification rules are simultaneously extracted. Experimental results show that BLUE produces models that are more comprehensible when compared with alternative methods. These experimental results lend support for future studies in visual data mining.

Acknowledgments

The last four years at Nova Southeastern University's (NSU) Graduate School of Computer and Information Sciences (SCIS) have been a challenging experience. I have learned a great deal and am indebted to many individuals who have made the success of this dissertation possible.

I especially want to thank my dissertation committee advisor, Dr. Junping Sun, for his great encouragement, support, enthusiasm, patience, and insight. He introduced me to the field of Knowledge Discovery in Databases, advised me, and gave me the latitude to pursue my interests. I am grateful to my committee members, Dr. Michael Laszlo and Dr. Lee Leitner, for their many helpful comments and invaluable advice. I am thankful to the faculty at SCIS' computer science department for providing me with the inspiration necessary to pursue research-based computer science topics. I also want to express my appreciation to the Dean, Dr. Edward Lieblein, for his support, encouragement, and emphasis on the importance of clearly articulating complex topics. In addition, the financial support of the California Institute of Technology administration is gratefully acknowledged.

Most importantly, I thank my wife, Karen, for her constant love, understanding, and support, for without which, this dissertation would not have been possible.

Table of Contents

Table of Contents
(continued)

Table of Contents
(continued)

List of Tables

Tables

List of Figures

Figures

List of Figures
(continued)

Chapter 1

Introduction

Motivation

Whether one is a scientist in a laboratory, an administrator in an office, or a mechanic in an automotive garage, databases are becoming inextricably linked with how one performs his/her respective profession. As the number and size of databases increase, the ability to extract useful knowledge in a reasonable time, with a minimum effort, and in a comprehensible format becomes an important issue. It is becoming difficult for organizations to manage the time and tools necessary to extract useful knowledge from these data sets (Agrawal, Imielinski, & Swami, 1993a).

The most common methodology used today to address this issue is to collect data over some period of time and then try to build models that represent relationships in the data sets. The purpose of building these models is twofold. One purpose is to help predict future events based upon past performance; another is to discover relationships in the data that were unknown before the model was constructed. Since many application domains are growing exponentially in data set size, the construction of these models is becoming increasingly difficult.

Researchers have addressed the need to analyze and interpret large data sets by developing several data mining techniques. One limitation associated with most of these techniques is their use of fully-automated information-theoretic or statistical algorithms.

These algorithms are typically non-interactive, hide the model derivation process from the user, require the assistance of a domain expert, are frequently application-specific, and do not always effectively and clearly translate detected relationships.

Data Mining

Database mining[1], sometimes called Knowledge Discovery in Databases (KDD), was created to aid with the discovery of unknown relationships among attribute values in data sets (Chen, Han, & Yu, 1996; Fayyad, Piatetsky-Shapiro, & Smyth, 1996; Han, Cai, & Cercone, 1992; Imielinski & Mannila, 1996; Piatetsky-Shapiro & Frawley, 1991b; Quinlan, 1993). Many techniques assist with the KDD process including machine learning, database querying, and statistics. Most systems utilize approaches that are based on automatic rule derivation algorithms and text-based expressions. These approaches do not represent data in a visually helpful manner. As a result, analysts have a difficult time making decisions based upon abstract and complex representations of data (Iizuka, Shiohara, Iizuka, & Isobe, 1998). Exacerbating the issue is the realization that only a portion of the data stored in a database is useful at any one time (Han, Chee, & Chiang, 1998). In an effort to increase the comprehensibility of a derived model, attributes are sometimes culled, however, the selection of which attributes are of most interest is a difficult one (Wu & Urpani, 1999). This predicament gives rise to an analysis of two successful data mining strategies and how they could be fused via a data visualization method to more flexibly address these issues. The first strategy involves

[1] Database querying is a deductive process. Axioms and theorems (i.e. JOIN, SELECT, or PROJECT) are combined in order to deduce other axioms and theorems (i.e. complex combinations of the aforementioned primitive database operators). The complex combination is deterministic. Database mining is inherently an inductive process. Regularities, or patterns, are *generalized* into a model representing the relationships

decision trees; the second involves a subset of association rules called classification rules.

Decision Trees

Data classification is an important aspect of data mining (Rastogi & Shim, 1998; Agrawal et al., 1993a; Mehta, Agrawal, & Rissanen, 1996). Classifiers analyze database tuples and place each tuple into a predetermined class. By assigning tuples to classes based upon its properties, the resulting model can be used later for prediction purposes (Quinlan, 1993).

A popular method employed for data classification utilizes a directed acyclic graph (DAG) called a decision tree. Decision trees have been utilized for representing knowledge since the mid-1960's (Hunt, Marin, & Stone, 1966). Many optimizations have been attempted through the years (Gehrke, Ramakrishnan, & Ganti, 1999).

Association Rules

In an effort to provide a more thorough representation of knowledge in databases Agrawal, Imielinski, and Swami (1993b) introduced association rules as an alternative to decision trees. A comprehensive association rule set represents every concept in that rule set. The process of determining all rules in a database is untenable for large and complex data sets (Klemettinen, Mannila, Ronkainen, Toivonen, & Verkamo, 1994; Liu, Hsu, & Ma, 1998; Piatetsky-Shapiro & Frawley, 1991b). Consequently, Agrawal et al. (1993a) proposed the use of templates to help guide the selection of relevant rules. This led to an analysis of the determination and detection of the "most interesting" rules.

---

in the data set. The derived model is valid only for the data set for which the model was constructed.

A number of approaches were taken to help with this process. Constraints to limit the untethered processing that would otherwise come about from a comprehensive induction of a complete rule set were considered (Klemettinen et al., 1994). Agrawal et al. (1993a) introduced support and confidence constraints that acted as a method of pre-pruning. Cendrowska (1987) introduced a modular rule mining system called PRISM that overcame some of the limitations decision trees possess when dealing with noisy data. Association rule induction has become an active field of research[2].

Data Visualization

According to Tufte (1990), the principles of information design date back as long ago as seven centuries. The evolution of these principles provides a foundation for computer data visualization.

Data visualization takes advantage of the human's ability to quickly and intuitively discern patterns and anomalies in images. These techniques promise a number of improvements in the area of data mining. With a properly designed interface, users would have the ability to obtain immediate feedback from database queries without worrying about the underlying data mining technique(s) being used on the back end (Iizuka et al., 1998). A clearly defined visualization methodology would enable users with the ability to view local detail of the underlying data but within a global context.

[2] The extraction of the "most interesting" rules usually refers to induction of strong rules, but rules which might be of little interest to the user (Silberschatz & Tuzhilin, 1996; Klemettinen et al., 1994).

Problem Statement and Goal

Most data mining methodologies derive rules through classification, association, or sequential induction methods (Agrawal et al., 1993a). These methods operate via preset rule derivation algorithms (Iizuka et al., 1998). With few exceptions, each approach attempts to automatically structure knowledge without guidance from the user. Many problems exist when knowledge is induced with these methods.

Statistical approaches have difficulty representing both large numbers of features as well as feature ranges (Emmons, Jennings, & Edwards, 1999). Rules induced from automated processes are not produced interactively; hence, rule induction cannot easily accommodate domain knowledge, expert knowledge, or prior knowledge. Consequently, these algorithms cannot backtrack or facilitate changes to the model. Finally, induced rule-sets tend to be so numerous that they cannot be reviewed tenably (Klemettinen et al., 1994; Liu et al., 1998; Piatetsky-Shapiro & Frawley, 1991b). A flexible and more optimized approach is possible by utilizing a new data visualization technique in conjunction with decision tree and classification rule machine learning techniques.

This paper proposes the development of an interactive visual data mining algorithm, BLUE, as an alternative to existing decision tree and classification rule algorithms. BLUE attempts to utilize the strengths associated with human sight and the visual framework of BLUE's Independence Diagrams (BIDS) to construct global models of data sets in the form of decision trees. Classification rules are simultaneously extracted from a given decision tree as it is grown. At any point in time, a user has the ability to backtrack to an earlier node and explore the data set for other possible patterns

that might exist. The framework from which BLUE was constructed entailed many goals; including:

1) BLUE could not be application-specific. It had to be able to manipulate a wide variety of data sets.
2) BLUE had to provide facilities for both domain-experts and non-domain-experts alike in guiding the extraction process.
3) The decision trees created with BLUE had to be comprehensible.
4) BLUE had to facilitate top-down model development.
5) Users had to have the ability to cull attributes before decision tree and rule extraction commenced.
6) A capability for backtracking had to be provided so that previously visited nodes could be revisited.
7) BLUE had to provide a means to visualize local detail between attribute pairs in a given relational database table as well as a global visualization method to provide a context upon which models could be derived.
8) BLUE had to provide a flexible means for selection of the initial splitting attribute of the decision tree. Justification for the selection had to be based upon experimental results.
9) Many classification techniques are based upon the induction of numerically valued attributes, however, many application domains consist primarily of categorically valued attributes. BLUE was fundamentally focused on categorically valued attributes.

Barriers and Issues

The construction of a visual data mining algorithm based upon two different machine learning methodologies posed many challenges. One issue was how to determine when one machine learning technique became more useful than the other. This led to an analysis of how data could be visually represented with each of the aforementioned approaches. If the two methods could be represented in a similar means to optimize the detection of correlation among attributes, the potential for discovering previously unknown knowledge would then be enhanced. The resultant data could then be viewed as a composite in one form. Consequently, one issue dealt with the coalescence of the two machine learning techniques in a format that took advantage of the strengths each technique possessed.

A number of other issues relevant to the specific machine learning techniques themselves had to be addressed. It is well known that decision trees that replicate sub-trees must at some point be analyzed with rules or more task specific attributes (Quinlan, 1993). Consequently, decision trees must clearly represent concepts; otherwise comprehension of the resultant model would subsequently become lost to the human observer. This occurs when too many nodes are expanded upon and/or groups of nodes are replicated. On the other hand, when association rules become incomprehensibly numerous, the overriding concept basis must be generalized (Han, Huang, Cercone, & Fu, 1996b). Rules that possess this problem must be subjected to a process that would define the hierarchical concept levels in a comprehensible form. At this point the concept basis might better be represented with decision trees.

Construction of the visualization methodology itself promised challenges. Existing visual data mining techniques typically utilize data cubes (Han et al., 1996a), circle segments (Ankerst, Keim, & Kriegel, 1996), and decision trees (Agrawal et al., 1993b) to represent data. These methods visually provide either local detail between attributes in a given database table or a global context of the entire table's contents, but rarely both. It is necessary to provide both views of the data if the user is to create a comprehensible resultant model that can be interactively explored.

Another issue that had to be addressed was the development of a software prototype. Imielinksi and Mannila (1996) point out that KDD querying tool development is one of the most difficult areas to address when constructing a system to perform knowledge discovery – namely, due to performance considerations. Addressing this area was expected to be a challenging segment of this dissertation topic.

Another barrier was the determination of which data sets should be used for testing purposes. The selection of specific data sets was based upon a number of issues:

1) A determination of which data sets had already been utilized to test the automated algorithm C4.5 (the most popular automated algorithm).
2) The data sets had to be selected with preference given to those that contain primarily categorical attributes.
3) The selected data sets had to be available in the public domain.
4) An analysis of which data sets could be tested in decision tree and classification rule formats.
5) Formats for presenting results had to be determined.

Limitations and Delimitations

A few limitations associated with the undertaking of this dissertation topic existed. Not all data sets contained categorical data, hence, a discretization of numeric attributes needed to take place. As a result, sub-par partitioning of some of the numeric attributes occurred. Also, a pruning algorithm was not developed for this study. Although the resultant models were comprehensible in nature, a simple pruning measure was used to cull resultant classification rules (see *Step 7* in chapter 3 for more detail). Consequently, the resultant number of induced rules was not always optimal.

BLUE addressed the following delimitations:

1) Data sets had to be presented in a flat-file format. This included attributes (independent variables) in the columns and tuples (separate instances) in the rows.
2) Testing had to include all tuples in a given data set.
3) All tuples had to possess the same dimensionality; missing values had to be either added during preprocessing or removed from the data set (see the appendices for details regarding each data set).
4) Attributes that contained only discrete categorical values, including linearly ordered sets, were considered. Textual attributes, such as name and address, were removed.
5) Classes representing the dependent variables could not be closely related.

Definition of Terms

Association – Association between attributes in a relational database table has traditionally been based upon measures of support and confidence. Those rules that meet a given support and confidence level are considered to be strong rules. Strong rules can provide an indication of how important a given concept may be, however, the given rule may not actually turn out to be "interesting" or "most important" due to subjective differences between users and the application domain that is under study. In an effort to better qualify an association as being "interesting" or "most important" there has been a call for the development of measures other than support and confidence (Silberschatz & Tuzhilin, 1996; Klemettinen et al., 1994). This dissertation uses classification rules to represent association in data sets. These rules were interactively extracted as decision tree nodes were extracted. This interactivity facilitated the extraction of interesting rules.

Binning – The process of converting continuous decision tree attributes to sets of discrete valued ranges.

Classification Tree – A decision tree that is based upon induction of categorically valued attributes.

Correlation – One way to qualify the relationship between two variables is with the use of correlation. Correlation allows one to estimate the strength of relationship between two variables with a single number. This number summarizes the linear relationship between two variables. A positive correlation indicates a positive sloping relationship between the variables while a negative correlation indicates a

negative sloping relationship between the variables. If the correlation is 0, it does not mean that the variables are not correlated, only that they are not correlated linearly. This paper utilizes BIDS (see *BLUE's Independence Diagrams (BIDS)* in chapter 3 for details) to view the correlation between attribute-pairs. In this context, correlation is a visual measure taking into account the size and color of correlation rectangles and their relationship to all other correlation rectangles in the current BIDS.

Correlation Rectangle – Each rectangle that is a part of an independence diagram, or BLUE's Independence Diagram (BIDS), is referred to as a correlation rectangle.

Currently Analyzed Set – The set of tuples currently under examination.

Dimensionality – The number of attributes in a given relation.

Goodness Measure – A measurement criteria that attempts to maximize a given measure; such as information gain.

Multivariate Partitioning – Unlike univariate partitioning, multivariate partitioning allows linear combinations of continuously valued attributes. Consequently, it allows for non-axis parallel partitions (sometimes called oblique decision boundaries).

Regression Tree – A decision tree that is based upon induction of continuously valued attributes.

Top Decision Node – Same as the root node in a decision tree. It can be thought of as the highest generalization level in the current decision tree.

Univariate Partitioning – A parallel-axis form of decision tree node splitting where only a one linear equation is considered at each split point.

Summary and Outline of Paper

This chapter described the motivating factors that precipitated the undertaking of this dissertation topic. Chapter 2 furnishes the reader with requisite background material on the KDD process in general as well as the origin, construction, and algorithmic approaches to existing decision tree and association/classification rule machine learning techniques. Fundamental classification algorithms are presented as well as issues that were considered in the development of the visual data mining algorithm called BLUE. Visualization techniques for representing data are also covered. Chapter 3 describes the methodology and theoretical foundations upon which BLUE was constructed. In addition, guidelines for reading BLUE's Independence Diagrams (BIDS) are presented as well as how they were incorporated into BLUE. Chapter 4 provides an analysis of BLUE's application to a number of data sets in two experiments. Chapter 5 provides an interpretation of the overall investigation undertaken in this dissertation. This includes conclusions, implications, as well as recommendations for future studies.

Chapter 2

Review of the Literature

Introduction

This chapter presents a review of some of the key elements that comprise the Knowledge Discovery in Databases (KDD) field in general and the sub-field of classification specifically. This includes many of the concepts, techniques, and algorithms utilized in today's decision tree and association rule classifiers. In addition, many data visualization techniques that help users identify relationships in data are reviewed.

Knowledge Discovery in Databases

The analysis of relationships in databases has been studied for many years (Michie, Spiegelhalter, & Taylor, 1994). Most analyses are based upon information theory or statistical metrics (Marmelstein, 1999). The overriding goal has been to discover useful, otherwise unknown relationships in the data, referred to as KDD (Chen et al., 1996; Fayyad et al., 1996; Han et al., 1992; Imielinksi & Mannila, 1996; Piatetsky-Shapiro & Frawley, 1991b).

The KDD process is composed of six generally defined steps. Each of these steps can be implemented with different techniques. Figure 1 exhibits the process. The basic steps are as follows:

1) Data Selection – Data sets can become very large and untenable to work with (Klemettinen et al., 1994; Liu et al., 1998; Piatetsky-Shapiro & Frawley, 1991b). Consequently, it is sometimes necessary to cull or randomly select a representative subset of the data to work with.
2) Cleaning/Preprocessing – The data selected in step 1 is processed by a preprocessing algorithm. This includes the translation of data into acceptable formats and addresses the issue of missing attribute values.
3) Transformation/Reduction – This step removes attributes from the data set that are not relevant to the problem domain. If it is determined that the inclusion of new attributes might facilitate the data mining process these attributes would be added at this step.
4) Data Mining – The data mining algorithm is applied to the data set from step 3. This includes the search for interesting patterns and the conversion of data into relevant representations for user interpretation.
5) Evaluation – It is important that a data mining algorithm produce the most relevant and/or useful information from a given database table. The evaluation step compares the output of step 4 to a type of "goodness measure" that assists in this determination.
6) Visualization – The visualization step transforms the resulting data set into a comprehensible form.

These six steps outline the general steps taken in knowledge discovery systems. Figure 1 includes double-ended arrows to indicate that the knowledge discovery process

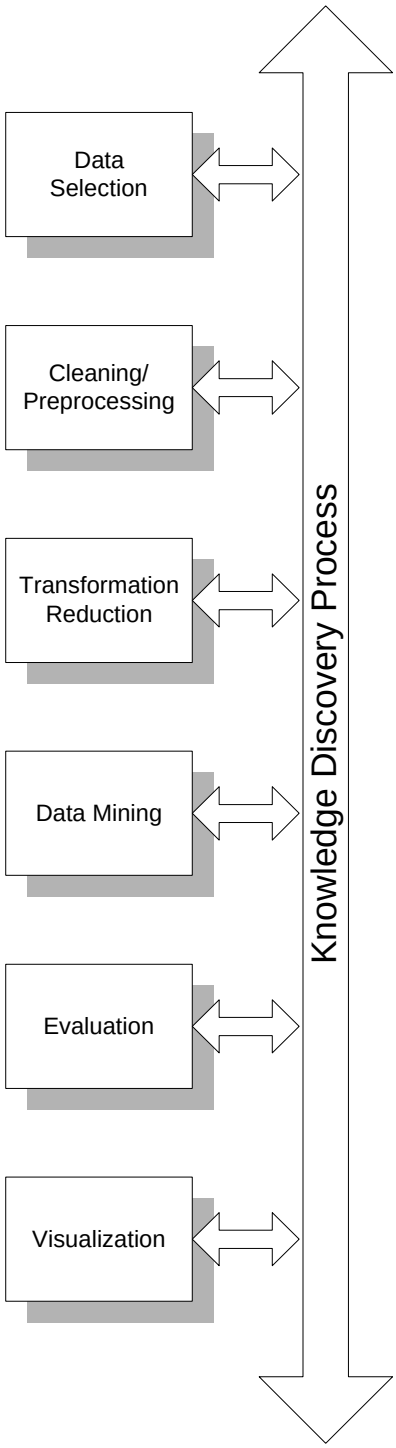


Figure 1: The Knowledge Discovery Process

is iterative. As new information is discovered, analysts revisit specific portions of the database in an effort to derive and interpret even more extensive information from them. In the context of this paper, the ultimate goal was to develop decision tree models that represent relevant independent variable(s) to dependent variable(s).

Decision Trees - Introduction

Decision trees are utilized for data classification. They are simple in construction and easy to comprehend. Consisting of nodes, branches, and leaf nodes (terminal nodes), these components are interconnected in the form of a directed acyclic graph (DAG). Each node represents a mathematical or logical test upon specific attributes in the data set. The goal is to unambiguously split (partition) the data set in a comprehensible way that ultimately represents the hierarchical interaction of variables. The outcome of each test determines how each node is induced. Parent nodes can have two or more child nodes, depending on the induction algorithm chosen. The parent and child nodes are connected via branches that represent the outcome of the test performed at the parent node. A leaf node has no children and corresponds to a class. Decision trees have zero or more nodes and one or more leaf nodes. An example decision tree is presented in Figure 2.

Historical Origins

Decision trees first gained acceptance via Hunt's et al. (1966) Concept Learning System (CLS). CLS uses a divide-and-conquer look-ahead partitioning method that

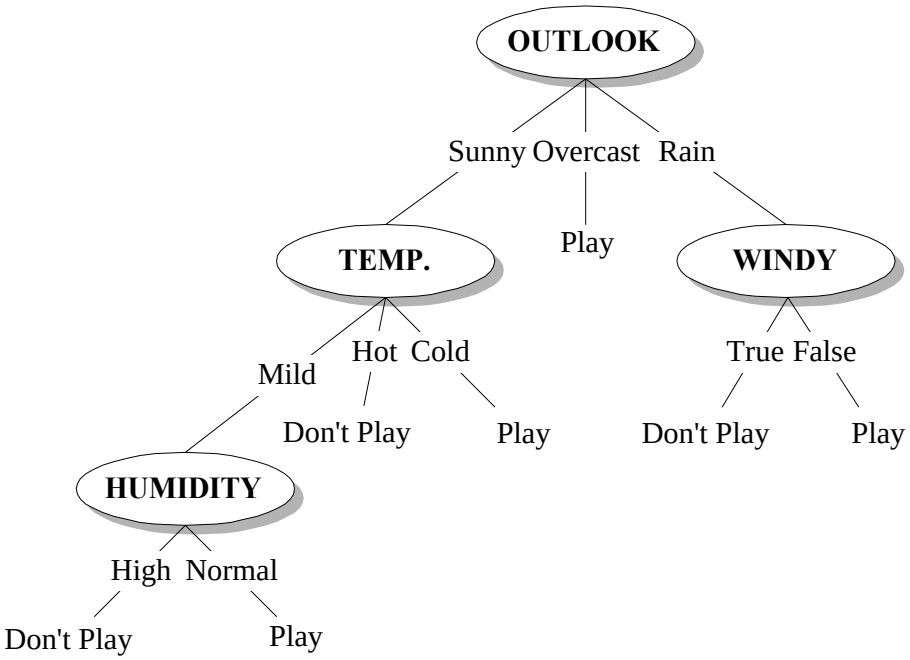


Figure 2: Saturday Morning Example

analyzes the construction of all decision trees at a given decision tree depth. Based upon the minimized cost of classifying a tuple, the appropriate decision tree is constructed. CLS is simple conceptually but expensive computationally.

The CLS approach can result in decision trees that perfectly classify the training data set. Difficulties arise, however, if the test data set contains noise; CLS does not possess a facility to handle this case. In addition, if the training data set does not contain a representative number of examples then the resultant model may not be supported statistically.

Friedman (1977) modified CLS to include a foundational algorithm that served as the basis for Classification And Regression Trees (CART) and Iterative Dichotomizer 3 (ID3). ID3's improvements over CLS included windowing and the application of a computationally inexpensive information-theoretic approach to selecting attributes (Quinlan, 1986). The idea was to minimize the entropy in a decision tree by minimizing redundant information, ultimately determining the smallest decision tree that fully described the training data set.

C4.5 (Quinlan, 1993) is the most popular and successful automated decision tree construction algorithm to date. C4.5's advancements over ID3 include the application of a *gain ratio criterion*. This criterion involves the calculation of information-gain normalized by an attribute's respective split-information. ID3 possesses a preference for splitting attributes that have many categories. The normalization through split-information minimizes the possibility of anomalous results.

Another successful approach is provided by Breiman, Friedman, Olshen, and Stone (1984). This approach, called CART, allows for the induction of multivariate

decision trees but is more computationally costly than ID3 or C4.5. CART utilizes the *Gini index* measure to calculate the probability of misclassification as opposed to information-gain's impurity of splitting attributes. This criterion overcomes ID3's bias for attributes with many outcomes by selecting attributes with the smallest *Gini index*.

Many other decision tree induction algorithms exist[3], however, most of them use a variation of one or more of the aforementioned algorithms. The focus in this dissertation was on C4.5 and its ancestors.

Benefits of Decision Trees

Decision trees possess many characteristics that make them good classifiers:

1) The decision tree formalism allows for the creation of trees that are intuitively simple and clear (Breiman et al., 1984; Mehta et al., 1996).

2) Knowledge acquisition via pre-classified examples allows for the construction of models without the direct consultation of domain experts. Learning from examples alleviates the *knowledge acquisition bottleneck* – it addresses the issue that even when domain experts are available it is difficult for them to express intuitive domain knowledge (Fayyad & Irani, 1992).

3) Exploratory analysis is facilitated because no parameters must be set before a decision tree is induced (Ankerst, 2000).

4) Decision tree induction is efficient at processing large training data sets (Gehrke et al., 1998; Mehta et al., 1996; Shafer, Agrawal, & Mehta, 1996;

[3] A short list includes QUEST,CHAID, FACT, SLIQ, RainForest and Sprint. RainForest and Sprint are

Fayyad & Irani, 1992). In addition, decision trees can be trained quickly, when compared with other classification techniques such as neural networks.

5) A large volume of data can be represented in a compact form (Murthy, 1997).

6) A split at a node in a decision tree is dependent on all prior splits leading up to that node. Consequently, attribute correlation is clearly identified and support and confidence measures can be applied to their outcome.

7) Decision trees are robust. They are insensitive to outliers and misclassification (Breiman et al., 1984).

8) The resulting decision tree is displayed in symbolic form that provides for a global context of the entire data set.

9) The resultant decision tree model can be converted to classification rules (Quinlan, 1993) and Structured Query Language (SQL) queries (Agrawal, Ghosh, Imielinski, Iyer, & Swami, 1992).

Drawbacks of Decision Trees

Decision trees suffer from some specific deficiencies, including:

1) There are cases where the simplicity of a decision tree might lead to misinterpretation. Consequently, if a decision tree attribute is never split it could be construed that this variable is not important and inadvertently

---

scalable classifiers.

removed – it may be the case that the attribute is masked by other variables (Breiman et al., 1984).

2) ID3, the traditional induction algorithm, operates by recursively splitting the training set into smaller examples. There comes a point when the splitting criterion is no longer supported statistically.

3) Decision trees do not always compactly represent Boolean concepts in Disjunctive Normal Form (DNF). Sometimes the "replication problem" occurs where sub-trees are duplicated within the same decision tree resulting in large, incomprehensible trees.

4) All examples from the training set are needed for the algorithm to function efficiently.

5) Splits in decision tree nodes are dependent on all prior splits leading up to that node. Once a split has been decided upon, it is never revisited.

While decision tree algorithms attempt to minimize tree size and maximize accuracy, these methods cannot use exhaustive search techniques due to computational complexity limitations in all but the most trivial cases (Safavian & Landgrebe, 1991). As a result, most decision tree algorithms use a greedy search criterion where splitting decisions are made node-by-node. Consequently, the search is effectively based on local optimization criteria and global considerations are not taken into account. In an effort to broaden the coverage associated with a local search methodology, researchers have attempted to modify the greedy search criteria with look-ahead algorithms (Marmelstein,

1999). These approaches have not been well accepted as the additional computational cost does not justify the benefits these algorithms offer (Murthy, 1997).

Decision Tree Induction

The process of inducing decision trees from data sets is the process of deriving specific conclusions from existing facts. This approach has been exemplified by research performed in the areas of machine learning, artificial intelligence, and pattern recognition. In the context of this paper, tuples were assigned to specific classes via a visual induction process. This process utilizes human sight to detect regularities and patterns within data sets. Based upon these facts, data is organized into finite, predetermined classes. The resulting decision tree represents a symbolic global representation of the data set.

Current decision tree induction techniques typically utilize either information-theoretic or statistical measures. These techniques induce decision trees via a two-step process. The first step is to grow the decision tree via a form of recursive partitioning. The second step is to prune the decision tree to minimize the possibility of overfitting the data and to provide a more comprehensible decision tree. Rastogi and Shim (1998) have found that in some cases, real-world data sets are pruned 90%, indicating a substantial waste of effort in the growing stage.

Partitioning

To construct decision trees that are concise, are accurate, and provide a facility for exploration, one must carefully consider attribute partitioning and splitting criteria.

These criteria determine when and how nodes are allocated in decision tree induction. The most popular partitioning approaches fall under two categories, univariate and multivariate (oblique) (Marmelstein, 1999).

Univariate partitioning entails a linear split of one attribute of the form

$$s_t \leq H \tag{2.1}$$

where s_t is the t^{th} splitting category and H is the classification attribute. In essence, univariate partitioning focuses on determining which attribute is the best discriminator at each node as the decision tree is grown. This approach provides a clear mechanism for attribute splitting and the corresponding rule-sets are comprehensible, however, univariate trees are sometimes needlessly large and complex because they are a parallel-axis form of partitioning (Dietterich, 1990; Marmelstein, 1999). Both ID3 and C4.5 utilize univariate partitioning.

Multivariate partitioning came about in an effort to create partitions that are based upon combinations of attributes (non-parallel axis). Multivariate partitioning is of the form

$$\sum_{t=1}^{d} b_t s_t \leq H \tag{2.2}$$

where b_t is the coefficient of the t^{th} attribute, s_t is the t^{th} splitting attribute, H is the classification attribute, and d is the number of dimensions in the data set. Algorithms based upon multivariate partitioning do produce smaller trees when compared with those produced with univariate algorithms, however, there is a computational cost associated with this approach that sometimes does not justify its implementation (Marmelstein, 1999).

Splitting Strategies

Once a partitioning strategy has been selected, the initial splitting attribute and its corresponding split points must be determined. Selection of the first splitting attribute defines the order in which the rest of the decision tree could possibly be induced. Possible sub-tree selection is dependent on prior parent nodes. The selection of one attribute instead of another attribute has been considered by a number of researchers (Fayyad & Irani, 1992; Quinlan, 1986; Ankerst et al., 1996; Breiman et al., 1984; Lewis, 1962). The selection process is most commonly based upon an impurity measure, entropy being the most popular.

Entropy/Information-Gain

Shannon's information gain theory (Shannon, 1948) is the foundation upon which ID3 and C4.5's splitting strategies are based. The theory, concerned with minimizing the cost of sending messages across noisy transmission lines, calculates the information conveyance of a possible message. If there are m equally weighted possible messages, the probability p that one of the messages will be selected for transmission is $1/m$. The information conveyed by this message is defined as $\log_2(m)$. Substituting p for $1/m$ gives $-\log_2(p)$. As an example, if there are eight possible transmission messages, the number of bits required to accommodate all eight messages is $\log_2(8) = 3$.

In most systems, more than one set of messages is transmitted across a given transmission line. Consequently, *Entropy*(P) is the total information that could possibly be transmitted across the given line. This equates to a summation of the probability distributions that could possibly utilize that line. Mathematically this can be expressed as

$$Entropy(P) = -\sum_{i=1}^{m} p_i \bullet \log_2(p_i) \quad (2.3)$$

where m is the number of message sets.

Entropy can be thought of as a measure of uncertainty, or randomness – the higher the entropy, the higher the uncertainty. As an example, consider a system that contains two sets of messages. The total information conveyed across a given transmission line can be determined by using equation 2.3. One set of messages will have probability p_1 and the other set of messages will have probability p_2. The total probability can be calculated as $p = p_1 + p_2$; p must sum to 1. Using this relation and equation 2.3, the total entropy for two sets of messages is given by

$$Entropy(P) = -[p_1 \bullet \log_2(p_1) + p_2 \bullet \log_2 (p_2)] \quad (2.4)$$

By rearranging $p_1 + p_2 = 1$, the equation can be rewritten as

$$Entropy(P) = -[p_1 \bullet \log_2(p_1) + (1 - p_1) \bullet \log_2 (1 - p_1)] \quad (2.5)$$

In the case where $p_1 = 1$ and $p_2 = 0$ there is no uncertainty, $Entropy(P) = 0$. If $p_1 = .67$ and $p_2 = .33$, $Entropy(P) = .91$. Finally, if $p_1 = p_2 = .5$, the $Entropy(P) = 1$. Maximal uncertainty occurs when $p_1 = p_2 = .5$; thus corresponding to the largest entropy value. This can be seen in Figure 3 where entropy is graphed as a function of message coverage.

In ID3 and C4.5, the goal is to determine which attributes should be selected as splitting nodes and in what order. The initial splitting node should be the one that maximizes information gain and minimizes uncertainty. Consequently, this selection corresponds to the subset of tuples that is the most homogeneous (Rastogi & Shim, 1998). Subsequent nodes are also selected using these criteria.

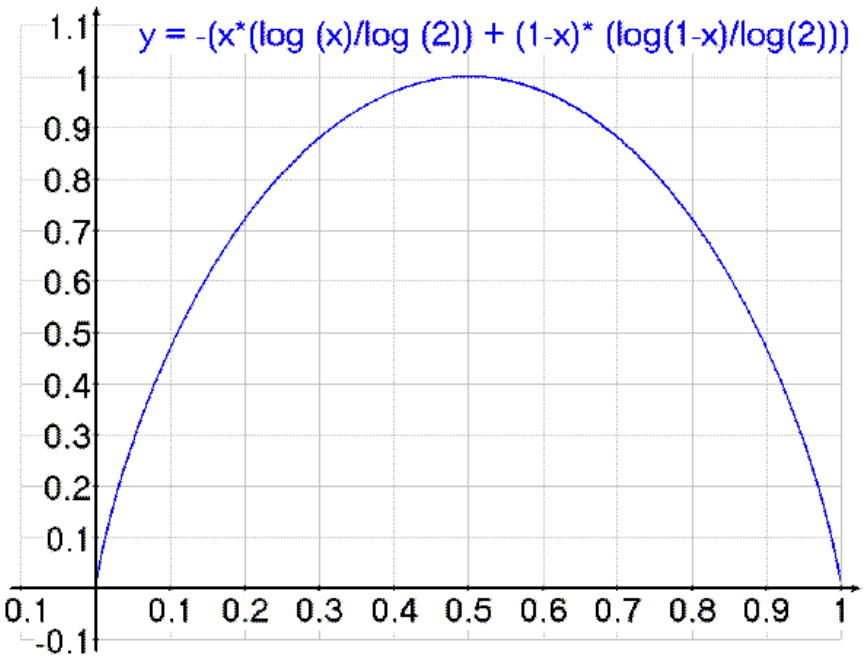


Figure 3: Percent of Message Coverage

Assume that a data set *S*, consists of a number of attributes A_1, A_2, … , A_n and a categorical classifying attribute *C* possessing *k* distinct classes. Further, assume that there are *T* tuples in the data set, each is partitioned into one of the *k* distinct classes of *C*. The information needed to correctly identify tuple class membership is

$$Entropy(T) = -\sum_{i=1}^{k} \frac{|C_i|}{|T|} \log_2 \left(\frac{|C_i|}{|T|} \right) \tag{2.6}$$

where *k* is the cardinality of *C*.

Entropy(T) results in a value that corresponds to the maximum information contained within *T*. In order to determine which attribute contributes the most to this value, the entropy of each attribute must be calculated and then compared. This can be accomplished by partitioning each non-classifying attribute A_1, A_2, … , A_n into sets of tuples T_1, T_2, … , T_m. The weighted average of the information needed to correctly identify the class of a tuple of T_i given attribute A_j can be represented by

$$Entropy(A_j, T) = -\sum_{i=1}^{m} \frac{|T_i|}{|T|} Entropy(T_i) \tag{2.7}$$

where T_i is the subset of data of *T* partitioned along attribute A_j's distinct categories.

With this information, the information-gain of each non-classifying attribute in *S* can then be determined by

$$InformationGain(A_j,T) = Entropy(T) - Entropy(A_j,T) \tag{2.8}$$

This equation represents the difference in the information needed to identify a tuple in *T* of class *C* and the information needed to identify a tuple in A_j of class *C* once it has been assigned to one of A_j's distinct categories. Another way to look at it is that it is the gain in information due to attribute A_j.

Once these values have been determined, it is possible to select the attribute with the highest information-gain as the initial splitting attribute. This procedure is recursively applied until a complete decision tree is constructed. The ultimate goal is to create decision trees that are small in size and meaningful in representation.

The information gain approach to decision tree induction is effective, however, it tends to create trees that are very large and are biased towards attributes that have many categories (White & Liu, 1994). For example, given an index attribute (an attribute that has different values for each element of the data set), its information gain is maximal when compared with other attributes that are not index attributes. Quinlan (1993) acknowledges this to be a problem and developed a method to normalize this bias. The normalization factor is referred to as the *gain ratio criterion*.

Gain Ratio Criterion

The *gain ratio criterion* relies on an analysis of the information involved with each split attribute. This is given by (Quinlan, 1993)

$$SplitInformation(A_j, T) = -\sum_{k=1}^{v} \frac{|T_k|}{|T|} \log_2 \left(\frac{|T_k|}{|T|} \right) \tag{2.9}$$

where v is the cardinality of attribute A_j. The gain ratio is calculated as follows (Quinlan, 1993)

$$GainRatio(A_j) = \frac{InformationGain(A_j, T)}{SplitInformation(A_j, T)} \tag{2.10}$$

The goal is to determine which classes differentiate themselves from other classes and simultaneously minimize the depth of the decision tree. This was facilitated by normalizing the information in attribute A_j via equation 2.9. Consequently, less detailed

splits are not unfairly penalized. However, if a given attribute has a predominance of one category, *SplitInformation* results in a high value, and *GainRatio* results in a low value. Consequently, anomalous findings could result.

Cardinality

Cardinality is a scalable classification and rule derivation technique that facilitates the induction of multi-dimensional databases (Sun, 2000). This method takes advantage of the fact that the attribute with the smallest cardinality for data partitioning possesses the largest support (see the section titled *Association Rule Construction and Optimization* for a definition of support and confidence).

The algorithm uses the intrinsic cardinalities of each attribute to build a model in a top-down manner. The algorithm recursively partitions the data set until support and confidence thresholds no longer meet a minimum criterion. Since each partitioned set is arrived at independently of the other partitioned sets, when one partition completes its partitioning, other partitioned portions of the tree can continue. This partitioning highlights the parallelism with which a cardinality-based splitting approach could be of advantage. Consequently, cardinality is an approach that efficiently determines attribute splitting order.

Personal Preference

The selection of initial splitting attribute is frequently independent of any measure applied to the data set such as entropy or support and confidence thresholds. These cases are considered to be research-based. The researcher possesses an interest in a specific

relationship among many attributes. By having the capability to direct the induction process the researcher is empowered with the ability to extract useful, interesting, and comprehensible information from the data set without following a pre-specified measure that might not allow the researcher to monitor specific interactions of the features – information-theoretic or statistical approaches do not offer this flexibility (see the *Summary* section in chapter 3 for further elaboration).

Guidelines for Decision Tree Growth

A key issue associated with the induction of decision trees is the determination of when to terminate the induction process. Mitchell (1997) has empirically determined that the accuracy associated with increasing the number of decision tree nodes first increases, levels off, and then decreases. This highlights the need to induce decision trees that are of a specific size. A number of approaches are available for determining when to stop the growth of a given decision tree.

Stopping Rules

Historically, stopping rules have possessed a dual definition. On one hand, they refer to the rules that define when the induction of a decision tree should be stopped. The other definition refers to early approaches researchers utilized to limit the growth of decision trees. These techniques relied on simple parameters to stop the induction of a decision tree before it was completely grown. For example, one approach limited the depth a given decision tree could grow. When the decision tree grew beyond this depth the induction process was stopped. Another approach utilized a threshold at each node

that kept a count of the number of records that might utilize the given node. If the number of records present at the node did not exceed the threshold value, the node was not added to the decision tree. Stopping rules suffer from decision trees that under-fit the training data (i.e. not adequately represent the data in the training cases).

Pruning

Stopping rules have largely been superceded by various pruning strategies, however, the guidelines that define specific pruning strategies are still commonly referred to as stopping rules (i.e. Breiman et al., (1984) for instance). Two common cases exemplify the two major pruning strategies. If the induction process is stopped too soon, then the representative tree might under-fit the data. If the induction process is not stopped soon enough, the representative tree might over-fit the data, inducing a data representation that classifies cases not represented in the training data set (i.e. anomalies). Pre-pruning and post-pruning represent the general categories of pruning utilized today (Breiman et al., 1984; Quinlan, 1993).

Pre-pruning strategies typically operate on a node-by-node basis. When a certain criteria is met, usually a statistical measure of support and/or confidence, node splitting along a given path is stopped. Consequently, the induction process proceeds with nodes along other paths through the decision tree. As with stopping rules, splitting is frequently stopped too soon, resulting in a tree that does not accurately classify records and underfits the data.

Post-pruning strategies allow for complete over-fitted decision trees to be produced. Decision trees are reduced in size by removing sub-trees that do not contribute

significantly to its classification accuracy[4]. Some common techniques include minimal cost complexity pruning, reduced error pruning, pessimistic pruning, and error-based pruning.

Frank and Witten (1998) determined that most pre-pruning techniques perform inadequately when compared with most post-pruning techniques. Consequently, most decision tree algorithms use post-pruning. A number of techniques have been utilized recently to induce trees of optimal size.

Minimum Description Length (MDL)

The Minimum Description Length (MDL) principle (Rissanen, 1978) has been applied to the machine learning field (Mehta, Rissanen, & Agrawal, 1995; Quinlan & Rivest, 1989) to make the determination of when to grow a decision tree node. MDL measures both the complexity and correctness of a model in a single evaluation of the data set. This contrasts with other methods that require separate training and testing data sets. Consequently, MDL is useful for small data sets that might be noisy and/or possess unevenly distributed training examples.

The MDL algorithm attempts to prune nodes where the encoding size has been minimized. If the number of bits required to compute the current message and the number of bits required to memorize the miscategorized examples exceeds the current description length, it does not continue to grow the decision tree. Consequently, MDL is useful for estimating the point at which the model begins to overfit the data.

[4] A model that perfectly classifies the training data may not result in the best generalized model. This is because the model may not accurately handle examples that were not available in the original training set.

Bagging and Boosting

Another form of decision tree classification uses a combination of multiple models and a voting criteria. Two popular methods are bagging (Breiman, 1996) and boosting (Schapire, 1990; Freund & Shapire, 1996). Both of these methods create multiple models based upon different samples from the training set.

Bagging reduces the generalization error for unstable learners (i.e. models). This is important in cases where the model would change drastically when training data is altered minimally. The algorithm works as follows. Given a training set size *n*, create *m* different training sets by sampling the original data set. Once the generalization error has been calculated for each *m*, the models are combined using a majority vote.

Boosting was developed by learning theorists looking for a way to guarantee performance improvements for weak learners. The algorithm begins by assigning equal weights to each tuple in a data set. With each iteration through the algorithm the weights assigned to the examples that were evaluated correctly are reduced. Likewise, the examples that were evaluated incorrectly are increased. The weighting can be thought of as a form of vote. Once a predetermined error rate has been reached, 0.5 for example, the algorithm stops. Bagging results in models that are consistently effective while boosting generally results in models that perform better.

Decision Tree Algorithms

The two most widely known decision tree algorithms, Quinlan's ID3 (Quinlan, 1986) and Quinlan's C4.5 (Quinlan, 1993), utilize the generic decision tree algorithm

shown in Figure 4. The principle difference between the two is in their respective partitioning and splitting strategies.

ID3

ID3 utilizes a splitting strategy based upon Shannon's information gain criteria (Shannon, 1948). More specifically, it focuses on top-down mutual information gain via the Shannon-Fano prefix coding scheme (Goodman & Smyth, 1988). This approach attempts to maximize the information gain at each node along an inductive path (see the section above, *Entropy/Information-Gain* for more information on this topic). ID3 is the precursor to C4.5.

C4.5

ID3's bias towards attributes that contain many values precipitated the creation of C4.5 (White & Liu, 1994) (see the section above, *Gain Ratio Criterion* for more information about this topic). C4.5 utilizes the information-theoretic approach outlined earlier, however, it utilizes the *gain ratio criteria* in order to reduce the aforementioned bias. In addition, rules can be easily extracted from decision trees grown with C4.5 by tracing each leaf node of the tree back to the root. Each path can be considered to represent a rule through the tree.

This is accomplished through the aid of a pessimistic pruning strategy. The resultant rule-set usually results in a simpler and more accurate classifier than that of the fully grown, overfitted, decision tree (Quinlan, 1987).

Algorithm: Generic Decision Tree Induction
Input:
i) database relation *D*
Output:
i) Induced decision tree
Method:
Step1: *Select task-relevant from D*
Step2: *Call procedure createDT.*
Procedure: *createDT*
BEGIN
IF all instances of *D* belong to the same class
THEN return
ELSE determine initial splitting attribute. Partition data set into subsets based upon attribute cardinality
FOR all child nodes
Call *createDT*
END {*createDT*}

Figure 4: Generic Decision Tree Algorithm

CART

Breiman's et al. (1984) CART is a multivariate decision tree induction algorithm. CART uses binary, or two-way, splits at each node of a decision tree. Consequently, the evaluation of a given expression posed at each node results in a true/false outcome. This approach facilitates the efficient storage of the induced model internally in the form of a binary tree. The process is recursive; it can be repeated by treating child nodes as parents.

CART utilizes a brute-force splitting technique. If there are Z tuples in a data set and Q attributes, CART will consider $Z*Q$ possible splits. These possible splits are rank-ordered based upon a goodness measure. The default goodness measure in CART is called the *Gini index*. The *Gini index* is a probability function that compares the probability for each split at each node. The probabilities are compared and the one that minimizes the misclassification at the given node is selected. The *Gini index* is

$$g(t) = 1 - \sum(p_i)^2 \tag{2.11}$$

where p_i is the probability of each class. The idea is that child nodes, and ultimately leaf nodes, are more pure than their corresponding parents.

Through the use of cross-fold validation CART prunes nodes. This technique determines if an increase in accuracy justifies the inclusion of an extra node. A maximal tree is grown and then sub-trees are derived from the maximal tree. Depending upon error rates, or estimated costs, the best sub-tree is selected – this may or may not result in the smallest tree.

The combination of CART's exhaustive search techniques and cross-fold validation techniques are effective but can be inefficient. CART can produce useful

results but can suffer from the inability to elude local optima. Consequently, CART can sometimes result in early node termination (Marmelstein, 1999).

Association Rule Construction and Optimization

Association rules represent one of the most popular forms of concept representation in machine learning. Introduced by Agrawal et al. (1993b), association rules possess a specific form. Consider a set of items $I = \{i_1, i_2,\ldots, i_n\}$ and a relation R that contains a set of database tuples U where $U \subseteq I$. If O is a set of items in I then a tuple in U, or U, is said to contain O if $O \subseteq U$. With this background, an association rule can then be defined as an implication of the form $O \rightarrow W$ where $O \subset I$, $W \subset I$, and $O \cap W = \varnothing$.

It is useful to utilize two measurement criteria to quantify the strength of rules created with this technique; namely, support and confidence. Given a proposed rule of the form $O \rightarrow W$, support s is defined as the percentage of tuples in R that satisfy $O \cup W$. Given the proposed rule $O \rightarrow W$, confidence is defined as the percentage of tuples in R that satisfy O that also satisfy W.

Agrawal and Srikant (1994) specify that rules extracted in this form do not need to be limited to those that contain only one consequent. The goal is to extract all rules that meet a specified minimum support and minimum confidence. A downside of this approach is that there is the possibility that an untenable number of rules could be induced from a given data set (Klemettinen et al., 1994; Liu et al., 1998; Piatetsky-Shapiro & Frawley, 1991b). Further, it may be found that many of the rules are trivial

and/or contradictory. Consequently, researchers have focused on how to induce only the "most interesting" rules (Agrawal & Srikant, 1995; Piatetsky-Shapiro, 1991a).

Rule extraction can be computationally expensive – strategies have been attempted to address the issue. The *apriori* property (Agrawal & Srikant, 1994) is an innovative approach. The *apriori* property states that every subset of a frequent itemset (i.e. an itemset that meets a minimum support level) must also be a frequent itemset. This constraint facilitated the use of the following strategy:

1) First iteration – identify just the frequent itemsets with one item
2) Subsequent iterations – extend the frequent itemsets from prior iterations to include another item. This generates larger candidate itemsets
3) Test all candidate itemsets to determine which ones are frequent
4) Repeat until no frequent itemsets are generated

Because the candidate sets are enlarged by the frequent itemsets, it can be assured that the optimization is correct. Figure 5 shows the *apriori* optimization algorithm as presented in Agrawal and Srikant (1994).

A number of efficiency improvements have been applied to *apriori* in recent years. One approach reduces the number of scans due to the fact that *k* database scans are made for *k*-itemsets with the standard *apriori* algorithm. Another approach utilizes indexing (hashing) to reduce the number of scans (Park, Chen, & Yu, 1995). Yet another approach is to reduce the size and number of transactions by applying heuristics.

Association rule inductive methods present many challenges. The sections presented above serve as an introduction to some of the issues that must be considered in

Algorithm: **Apriori**
Input:
i. Database, D
ii. Minimum support threshold, *minsupp*
Output:
i. L_k, set of large itemsets that fulfill *minsupp*
Method:

L_1 = {large 1-itemsets};
for ($k = 2$; $L_{k-1} \neq 0$; k++) do begin
  C_k = **apriori-gen**(L_{k-1}); //set of new candidates
  forall ($t \in D$) do begin // support counts
    C_t = subset(C_k, t); // candidates in t
    forall ($c \in C_t$) do // for all candidates
      $c.count$++;
  end
  $L_k = \{c \in C_k \mid c.count \geq minsupp\}$
end
Answer = $\cup_k L_k$;

where:
C_k = set of candidate itemsets (potentially large)

procedure **apriori-gen**(L_{k-1}) //determines superset of all large k-itemsets
insert into C_k
select p.item$_1$, p.item$_2$,…, p.item$_{k-1}$, q.item$_{k-1}$
from $L_{k-1}p$, $L_{k-1}q$
where p.item$_1$ = q.item$_1$,…, p.item$_{k-2}$ = q.item$_{k-2}$, p.item$_{k-1}$ < q.item$_{k-1}$
// now the pruning step
forall itemsets $c \in C_k$ do
  forall (k-1) subsets s of c do
    if ($s \notin L_{k-1}$) then
      delete c from C_k
return C_k;

Figure 5: *Apriori* Optimization Algorithm from Agrawal and Srikant (1994)

order to design algorithms that induce rules. Other areas include the application of constraints to the database tables before rules are extracted. Some interesting approaches can be found in (Agrawal et al., 1993a; Agrawal et al., 1993b; Agrawal & Swami, 1994; Bayardo Jr. & Agrawal, 1999; Frank & Witten, 1998; Han et al., 1992; Wu & Urpani, 1999). In addition, an active area of research involves limiting the number of rules that can be induced in a given data set during an induction run.

Benefits of Association Rules

Conceptual examples from a grocery store database presented in Agrawal et al. (1993a) highlight some of the benefits of induced rules.

1) "Find all rules that have diet coke in the consequent" – this type of association rule could allow analysts to determine what products could be put on sale to increase the sale of diet coke.
2) "Find all rules with bagels in the antecedent" – this type of association rule could appraise analysts of what product sales might be affected by the discontinuation of bagels.
3) "Find all rules with sausage in the antecedent and mustard in the consequent" – this type of association rule could help determine what products could be sold along with sausage to increase the sale of mustard.
4) "Find the best k-rules with bagel in the consequent" – this approach could cull rules to a tenable size for analysts to deal with.

Many times, the most interesting rules are those that come about unexpectedly. Discovering interesting rules from data sets has been considered by a number of

researchers (Klemettinen et al., 1994; Piatetsky-Shapiro, 1991a). The prime motivating factor for performing this research has to do with the large quantity of rules that can otherwise be induced.

Drawbacks of Association Rules

Association rules suffer from some specific deficiencies. These include:

1) The number of rules that can be induced from a data set can be enormous. Consequently, the number of rules that can be extracted becomes overwhelming (Klemettinen et al., 1994; Liu et al., 1998; Piatetsky-Shapiro, 1991a).
2) Once the rules have been induced, it is not easy to present them visually. The analysis of induced rules is problematic if they are presented only in the form of an implication.
3) Exhaustive rule induction results in many trivial and contradictory rules.

Data Visualization

Computer data was first commonly represented with text. The invention of the spreadsheet allowed for dimensional relationships to be displayed in tabular form. Line graphs, bar graphs, pie charts, and area charts were eventually added as visual aids. Other methods were developed in an attempt to clearly represent data. These techniques included scatterplots and histograms (Iizuka et al., 1998). As time moved on, an emphasis from the textual to visual form naturally followed (Shneiderman, 1994). The enabling factor that facilitated this progression was the human visual system.

The human visual system possesses several characteristics that make it useful in KDD exploration. Pattern recognition abilities for identifying interesting patterns, anomalies, and outliers are superior to other existing synthetic technologies (Cox, Eick, Wills, & Brachman, 1997). The ability to perform rapid visual recognition and provide comprehensible understanding relationships is naturally facilitated (Gershon & Eick, 1998). Further, Agrawal et al. (1993a) purport that the iterative discovery process is best facilitated via human input.

Data visualization systems have been developed to aid in the decision-making process (Iizuka et al., 1998). Some methods provide a global view of data at the expense of local detail (Jerding & Stasko, 1998). Other systems (Ankerst, Elsen, Ester, & Kriegel, 1999; Sarkar & Brown, 1992) utilize methods that allow for the presentation of local detail but also offer a global context upon which decisions can be facilitated.

It is interesting to note that there has been an emphasis in graphical techniques that utilize numeric data rather than graphical techniques that utilize categorical data. One explanation is that categorical data must be represented with a new graphical metaphor; one that is different than that used for numeric data. Another explanation is that categorical graphical approaches tend to be application-specific (Friendly, 1995).

A number of data visualization methodologies are presented below that support, to differing degrees, both numeric and categorical data. It can be seen that most of these methods differ based upon the number of dimensions that must be presented on a computer monitor at one time. A general synopsis of graphical data representation can be found in Fortner (1995).

Linegraphs and Bar Charts

These methods are used most commonly to represent data in one dimension. Linegraphs work well if multiple one-dimensional data sets must be compared against one another. This approach does not work well for bar charts. Bar charts are useful for small sets of data.

Scatterplots and Parametric Plots

These methods work well for viewing data in two and three dimensions. Scatterplots place data on the screen by using an axis as a function of another axis. Sample points are typically colored with a specific value relating to another variable such as temperature, humidity, etc. Parametric plots are scatterplots with lines drawn between the graphic values.

Greater than 3-Dimensions

Scatter matrix plots are useful for graphing more than three dimensions of data onto a two-dimensional plot. They work by graphing each dimension of the data set with all other dimensions of the data set. This approach is good for analyzing highly correlated data.

Visual Spreadsheets

Spreadsheets have been around for a number of years. The capabilities associated with horizontal and vertical matrices of data are easily manipulated for basic addition, subtraction, and multiplication operations due to its direct manipulation interface. This

makes the data entry method comprehensible for novice and expert users alike. The method utilized in Chi, Riedl, Barry, and Konstan (1998) takes advantage of these techniques while simultaneously allowing cells to contain visual objects.

Histograms

Histograms are graph-like structures used to represent the number of data point occurrences that fall within a series of ranges, or bins. They are used to evaluate data that occur at a certain frequency. The graphical representation is a bar graph.

The Information Mural

Jerding and Stasko (1998) describe a data visualization system called the Information Mural that displays a global view of an entire data set – this provides context for more detailed views. They describe the importance of having a global view of a data set for navigational purposes as well as for data analysis.

Fisheye Lens

A graphical fisheye view displayed on a computer screen is similar to an optical fisheye view utilized via a camera lens. Objects that are in the direct view of the lens appear magnified whereas objects that appear off-center of the lens are reduced in size. Sarkar and Brown (1992) present a method that allows one to view local detail of a data set without losing focus of the global context. This is important as viewers are less likely to lose the context upon which they are viewing the data set[5].

[5] The impetus for the work into the fisheye lens has its origins in Furnas (1982).

Circle Segments

This is a pixel-per-value visualization technique that utilizes one colored pixel per data point. A circle is split into segments. Each segment corresponds to a dimension in the data set. Color is utilized to view clusters in the data. Ankerst et al. (1999) develop a framework for deriving decision trees from circle segments from databases that contain numeric attributes.

Independence Diagrams

Berchtold, Jagadish, and Ross (1998) invented a data visualization technique that represents two dimensions of a relational data set with something called an independence diagram. The benefit of this approach is that it allows for a proportioned view of attribute pairs. Independence diagrams are not sensitive to outliers or other anomalies. This method could be used to visually classify data sets by novices as well as domain experts given clearly defined guidelines for viewing the independence diagrams.

Contribution

The goal of this dissertation was to create an interactive visual data mining algorithm, BLUE, that extracts decision trees and their associated classification rules. This work makes the following contributions:

1) Its proposed methodology simultaneously extracts classification rules and constructs decision trees from a new type of independence diagram called BLUE's Independence Diagrams (BIDS). The strengths decision trees possess and the strengths classification rules possess in inducing,

classifying, and representing knowledge is combined in a visual framework that uses BIDS to analyze local detail between attribute pairs and decision trees to provide for a global context of the entire data set. This approach results in more expressive models.

2) Its attribute-oriented approach facilitates top-down model creation. A predetermined splitting strategy is not required to utilize BLUE, however, if a user would want to use a specific splitting strategy, this could be accommodated.

3) The number of independence diagrams that Berchtold et al. (1998) recommend users analyze at any one time is $n(n\text{-}1)/2$. The induction methodology presented in this paper reduces this number to $(n\text{-}1)$ (see *Image Reduction Strategy* in chapter 3 for details).

4) Users are able to interactively guide the induction and exploration of a given data set. This enables backtracking and the ability to explore various combinations of attributes that could not be analyzed with information-theoretic or statistical methods alone.

5) BLUE was designed to support the induction of categorically valued attributes. This facilitates its use in the analysis of many real-world applications.

6) Guidelines were developed with the intent for both domain and non-domain experts to effectively induce decision trees and their corresponding classification rules.

7) Classification rules are extracted as the decision tree is grown. This provides for an alternative means of concept representation that could be easily represented in a computer.

Summary

This chapter introduced decision tree classification, association rules, classification rules, and highlighted a number of strengths and weaknesses associated with each method. In addition, some of the more popular data visualization methods were presented. Decision trees, classification rules, and independence diagrams were selected for further study because their combined strengths complement each other well. In addition, many of today's practical applications are facilitated through the use of decision trees and association rules (Liu et al., 1998).

The convergence of decision trees, classification rules, and a data visualization methodology facilitates the creation of comprehensible models, however, having a data visualization method alone does not facilitate visual knowledge discovery. An effective knowledge visualization system must include data visualization methods that provide local detail within a global context, a distortion technique to explore various relationships in the data set, and an interaction technique to guide the induction process.

Chapter 3 outlines the methodology that was used in the development of BLUE. The problem is explicitly stated and the problem solution articulated. BLUE is further defined and delineated from other decision tree and association/classification rule induction techniques. BIDS are further outlined as well as guidelines for how to read

them. A test methodology is presented outlining how BLUE was empirically tested. In addition, a description of format results is provided.

Chapter 3

Methodology

Introduction

This chapter outlines the research methodologies that were employed for the development of the visual data mining algorithm called BLUE. Step 1 defines a high-level framework for the design and implementation of the target algorithm. Step 2 establishes guidelines for the use of BLUE's Independence Diagrams (BIDS). Step 3 develops a software prototype that allows for the display and analysis of BIDS. Step 4 defines BLUE's algorithmic operation. Step 5 specifies the data sets that were utilized for experimental testing purposes. Step 6 outlines BLUE's reliability and validity. Step 7 specifies explicit testing procedures for the data sets in step 5.

Step 1 – Approach

The methods outlined in this chapter allowed BLUE to be evaluated based upon two proficiencies: 1) decision tree size/number of classification rules induced, and 2) comprehensibility of the resultant model. The first proficiency can be measured in quantifiable terms whereas the second proficiency can only be measured in qualitative terms[6]. Even though decision tree size and number of rules induced are quantitative

[6] Silberschatz & Tuzhilin (1996) differentiate interesting pattern detection in knowledge discovery systems into objective measures and subjective measures. Klemettinen et al. (1994) and Piatetsky-Shapiro &

measures, the impetus for creating BLUE was for the creation of a visual data mining approach that relied on user's visual selectivity preferences and knowledge of the data set (or lack thereof) to guide the induction process. The decision trees created in this dissertation were fully grown, overfitted models. A basic pruning stopping rule was utilized to prune the resultant decision trees so that they could be evaluated based upon the statistical measure of support.

BLUE was developed iteratively. Guidelines were established to utilize independence diagrams for viewing local detail in attribute pairs in relational databases. Multiple attribute selection strategies were considered to determine if and when a given attribute should be split. A decision tree induction methodology was created to provide a global context upon which individual BIDS could be analyzed[7]. A procedure was developed to monitor and direct attribute and tuple usage.

BLUE advances current data mining practice in a number of ways. BIDS are utilized for viewing local correlation relationships between attribute-pairs in relational data sets. In addition, BIDS support the simultaneous creation of decision trees and their corresponding classification rules. Decision trees provide users with a global context of the data set. This visualization framework provides domain and non-domain users with the ability to interactively guide the induction process. BLUE's methodology encompasses the extraction of knowledge from databases via data visualization. It was desired that this approach would allow users a means to flexibly explore a data set.

Matheus (1994) provide arguments for the need for subjective measures in knowledge discovery systems.

[7] Furnas (1982) provides insight into the importance of a visualization system that provides local detail within a global context. The idea, introduced as a conjecture in Furnas' paper, is that by providing such a system users can view structures they are most interested in.

Consequently, researchers would possibly have a manageable tool that could be used to better understand the underlying relationships in data sets while at the same time non-domain experts[8] would have specific guidelines to direct the induction process.

Step 2 – Independence Diagram Analysis, Framework, and Guidelines

Berchtold et al. (1998) demonstrate how independence diagrams can be used to display complex relationships between attribute pairs in relational data sets. The authors note that through the detection of "interesting" rectangles, statistical significance values and simple rules can be inferred. What is still needed are guidelines that address the following issues:

1) A methodology for extracting rules.

2) How to determine the most interesting rules.

3) How classification or prediction could be facilitated.

4) How decision trees could be induced by utilizing independence diagrams as the guiding visualization method.

5) How to handle large numbers of attributes.

6) How to provide for a global context upon which the local detail afforded by independence diagrams could be monitored.

7) How supervised learning could be supported.

Without such guidelines, the reader must use his or her own discretion to define a framework from which independence diagrams could be used in practice, leading to

[8] In the context of this paper, BLUE can be thought of as utilizing supervised learning – sometimes referred to as learning from examples.

potentially inconsistent results. The guidelines listed above served as a foundation upon which to establish a unified approach.

Independence Diagrams – Introduction

This section introduces guidelines for reading and interpreting the type of independence diagrams utilized in this dissertation, called BLUE's Independence Diagrams (BIDS). These guidelines are unique to BLUE due to its emphasis on combining decision tree and classification rule machine learning techniques including its emphasis on categorical attributes. This framework was developed based upon simplicity, ease-of-use, and comprehensibility of the resulting model. Models developed with BLUE were interactively created subsequently enhancing comprehensibility. The guidelines established below should be considered valid for the selected data sets tested in this paper and not generalizable for all data sets.

Independence diagrams are two-dimensional images that compare two attributes from a relational database relation. The image format is a combination equi-depth and equi-width histogram, referred to as an equi-slice histogram. Two types of measures can be obtained from independence diagrams. The first measure is a visual magnitude comparison of two attributes. The second measure is a grayscale representation of the density of tuples resident within independence diagram correlation rectangles (i.e. the rectangles that comprise the independence diagrams). The resulting diagrams facilitate user detection of interesting patterns in data sets that would otherwise go without detection. These general ideals were utilized in this paper to develop a form of

independence diagram that could be used for the interactive induction of decision trees, and extraction of their classification rules, that primarily contain categorical attributes.

The framework developed in this dissertation serves two purposes. One purpose is to provide researchers with a tool that could help with the interactive exploration of relational data sets. The second purpose is to provide users, either domain experts or non-domain experts, with the ability to develop comprehensible models that could be used for prediction and/or classification purposes.

The data sets utilized in this dissertation were presented to BLUE in standard relational form. Each data set encompassed a number of columns of data, referred to as attributes or features, and a number of rows of data, called tuples or vectors. Each data set consisted of a dependent variable, called the classification attribute. The classification attribute was comprised of a number of classes from which tuples in the data set could possibly belong. The remaining attributes represented the independent variables from which possible concepts could be extracted.

Independence Diagrams – Image Granularity

Berchtold et al. (1998) view and compare independence diagrams in three steps: 1) create independence diagrams for every attribute-pair combination, 2) view these diagrams as thumbnails, and 3) enlarge and analyze independence diagrams that seem interesting. It was decided to implement a similar strategy in this dissertation. This required considering the number of BIDS that should be displayed at one time.

To account for all possible combinations of independence diagrams for a given relational database table, a recursive relationship on the order of $V_n = V_{n\text{-}1} + (n\text{-}1)$ is

necessary, where V_n is the number of independence diagrams associated with the table's n attributes. This relationship is explicitly represented by $V_n = n(n\text{-}1)/2$. If a data set possessed 5 attributes, 10 independence diagrams would have been needed for an analysis. When n = 10, 45 independence diagrams would have been needed; when n = 15, 105 independence diagrams would have been needed. It is not uncommon for database tables to contain hundreds of attributes. For such large data sets, it is difficult to compare and contrast all images at one time on a single Cathode Ray Tube (CRT). Hence, large data sets impede image comprehension when utilizing this strategy.

Subsequently, an alternative strategy for viewing and comparing BIDS was chosen. Before a BIDS was created, a classification attribute was selected. The next step was to determine the initial splitting attribute. BLUE, as tested in this paper, accommodated gainratio, cardinality, and personal preference splitting strategies. Once the first node was determined, the splitting attribute was compared against the classification attribute via a BIDS to see if any direct correlation existed. If a correlation did exist, a leaf node(s) and its corresponding classification rule(s) were immediately extracted. The tuples corresponding to the identified correlation were then removed from the data set and the remaining attributes were arranged by class and compared against the classification attribute.

Image Reduction Strategy

The approach taken in Berchtold et al. (1998) generates an unordered arrangement of independence diagrams for all combinations of attributes (i.e. $n(n\text{-}1)/2$ where n represents the number of attributes in the given database table). While useful for

analyzing all possible dependencies between attribute-pairs, it is inefficient for the problem undertaken in this dissertation, namely, classification. This inefficiency meant analyzing every attribute-pair combination was not desirable, only those associated with the classification attribute. The number of BIDS required for viewing could then be determined through the following reasoning.

First, the question was asked, given n attributes in a relational database table, how many combinations of attributes must be made to evaluate each attribute with the class attribute only once? It can be seen that if n_c represents the classification attribute then there are n-n_c remaining attributes in the table. Since there can only be one attribute designated the classification attribute n_c can be replaced with 1 indicating that there are n-1 other attributes in the table. It follows that each of these n-1 attributes can be compared with the class attribute only once, hence, n-1 BIDS were required for comparison purposes in this paper. By utilizing this approach, the number of BIDS that had to be analyzed was reduced from $n(n\text{-}1)/2$ to $(n\text{-}1)$.

Independence Diagrams – Learning Strategy

The subject of learning from examples has interested researchers for many years. The primary appeal of this approach has been to determine if it can overcome what has been called the knowledge acquisition bottleneck. This bottleneck occurs because frequently expert-users have difficulty in expressing concepts clearly and concisely. Learning from examples utilizes data from pre-classified examples in an effort to minimize this bottleneck (Fayyad & Irani, 1992).

Independence diagrams utilize a form of unsupervised learning[9]. The algorithm processes the data set and corresponding pixel mapping locations are determined. This mapping is utilized for displaying corresponding grayscale pixel values. The given display technique in essence provides local detail of two attributes in a data set but a global learning methodology among various attributes is not provided.

BLUE's use of independence diagrams focuses on single and multiple class supervised learning. At each node of a decision tree, the induction of the next node is directed by the user. In the case of single class learning, the classification attribute consists only of two values, named the positive and negative instances (Quinlan, 1986). Multiple class learning discriminates among many classes. Each tuple in a given data set can belong to only one class.

BLUE's Independence Diagrams (BIDS)

This section establishes guidelines for reading the types of independence diagrams used for the induction of models utilizing BLUE. These diagrams differ from the independence diagrams found in Berchtold et al. (1998) in a number of ways including the inclusion of color, an alternative tuple density measure, and a focus on the extraction of categorical attributes. Consequently, BLUE's independence diagrams are referred to as BIDS (BLUE's Independence Diagrams).

The first implementation detail that differs from independence diagrams is the addition of color. Independence diagrams utilize a grayscale to represent the rectangles (called correlation rectangles in this dissertation) that make up the independence

[9] Independence diagrams are constructed via preset bucket width and bucket depth parameters. If either of

diagrams. Preliminary experiments indicated that by using a grayscale to detect correlation between attribute pairs it was difficult to determine when two correlation rectangles were fully correlated or not. Consequently, color was introduced to better discern similarities and differences between correlation rectangles. Green was used to show the correlation rectangles that have the highest tuple population within given BIDS. Red was utilized to show which correlation rectangles have no tuple population. Correlation rectangles which contain a population between the two extremes were linearly interpolated with a grayscale (i.e. each correlation rectangle was assigned a specific grayscale value based upon its respective tuple population).

BIDS also differ in the calculation of tuple density. Berchtold et al. (1998) determined that counting tuples within each correlation rectangle, sorting those counts, and then applying white to the highest count and black to the lowest count yields mostly black images. Consequently, the authors utilize a 5% and 95% quantile count as the darkest and lightest values respectively. By utilizing color, as outlined above, it was expected that quantiles would not be necessary.

In order to specify visualization guidelines for reading BIDS, definitions for the environment under which BLUE can be utilized are outlined below:

Definitions

Definition 3.1 BLUE facilitates the visualization of attribute pairs from databases represented in standard relational form.

these parameters are exceeded a new bucket is created – exceptions can be made to the bucket boundaries.

Definition 3.2 One attribute in a given database table must be defined as the classification attribute. Let the classification attribute C be the set of i classes $\{C_1, C_2,\ldots, C_i\}$ whose domain $dom(C_i)$ refers to the set of all possible categories for attribute C. This is represented by

$$C = \sum_{i=1}^{N} C_i \tag{3.1}$$

where N is the cardinality of the classification attribute. For example, in Table 1 attribute *PLAY* has been selected as the classification attribute. It follows from equation 3.1 that $C = \{Play, Don't\ Play\}$.

Definition 3.3 Individual class attribute categories correspond to a set of tuples that satisfy a class condition. This is represented by

$$C_i = \{t \in T \mid cond_i(t)\} \tag{3.2}$$

where t is a tuple from the set of tuples T. $cond_i$ refers to a tuple whose attribute categories fulfill a classification condition.

Definition 3.4 Tuples represented in a BIDS should sum, in both vertical and horizontal directions, to the number of tuples in the currently analyzed set. This is expressed by

$$|T| = |T_h| = |T_v| \tag{3.3}$$

where $|T|$ is the total number of tuples in the currently analyzed set, $|T_h|$ is the total number of tuples in the horizontal direction of a BIDS, and $|T_v|$ is the total number of tuples in the vertical direction of a BIDS. $|T_h|$ is given by

$$|T_h| = \sum_{i=1}^{N} |t_i| \tag{3.4}$$

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
Sunny	Hot	High	False	Don't Play
Sunny	Hot	High	True	Don't Play
Overcast	Hot	High	False	Play
Rain	Mild	High	False	Play
Rain	Cool	Normal	False	Play
Rain	Cool	Normal	True	Don't Play
Overcast	Cool	Normal	True	Play
Sunny	Mild	High	False	Don't Play
Sunny	Cool	Normal	False	Play
Rain	Mild	Normal	False	Play
Sunny	Mild	Normal	True	Play
Overcast	Mild	High	True	Play
Overcast	Hot	Normal	False	Play
Rain	Mild	High	True	Don’t Play

Table 1: Saturday Morning Relation

where N is the cardinality of attribute T_h and t_i is the number of tuples given for each category of T_h. $|T_v|$ is given by

$$|T_v| = \sum_{j=1}^{N} |t_j| \tag{3.5}$$

where N is the cardinality of attribute T_v and t_j is the number of tuples given for each category of T_v.

Definition 3.5 The summation of the number of tuples resident in all correlation rectangles should be equivalent to the total number of tuples in the currently analyzed set. This is expressed by

$$|T| = \sum_{i=0}^{Y-1} \sum_{j=0}^{X-1} r_{ij} \tag{3.6}$$

where $|T|$ is the total number of tuples in the currently analyzed set, Y is the number of correlation rectangles along the y-axis, X is the number of correlation rectangles along the x-axis, and r_{ij} is the number of tuples in the current correlation rectangle.

Definition 3.6 Tuple density is calculated through the summation of tuples resident within a given correlation rectangle. The highest density correlation rectangle is assigned the color green. If no tuples are resident within a given correlation rectangle, it is assigned the color red. All other correlation rectangles are assigned a grayscale value linearly interpolated from these extremes.

Definition 3.7 A fully correlated BIDS consists of green, red, and grayscale correlation rectangles that unambiguously place tuples in predetermined classes. An example of a fully correlated BIDS can be seen in Figure 10b.

Singular and Multitudinal Image Comprehension

A user versed in reading BIDS will encounter one of two categories of BIDS. These categories consist of singular cases, those consisting of single BIDS; and multitudinal cases, those consisting of many BIDs that must be compared. Singular cases are experienced during the selection of the root or leaf nodes of a decision tree. Multitudinal cases are experienced during the selection of internal nodes of a decision tree or to determine the initial splitting attribute.

A singular case representing the root node in a decision tree is shown in Figure 6. In this case the goal was to determine if any of the categories of the *y*-axis were fully correlated with any of the classes of the classification attribute – represented in the *x*-axis. If a category was fully correlated it meant that for all tuples of a given category, all tuples were fully classified. The red and green rectangles indicate a full correlation between a category in the *y*-axis attribute and the *x*-axis attribute[10] [11]. A leaf node can be derived from the root node *y*-axis correlation rectangle Y_{red} and the right *x*-axis classification column X_{right}. Simultaneously, a rule can be extracted $Y_{red} \rightarrow X_{right}$. The resultant decision tree and extracted rule are shown in Figure 7.

[10] This dissertation utilizes the *x*-axis to represent the classification attribute.
[11] See definition 3.6.

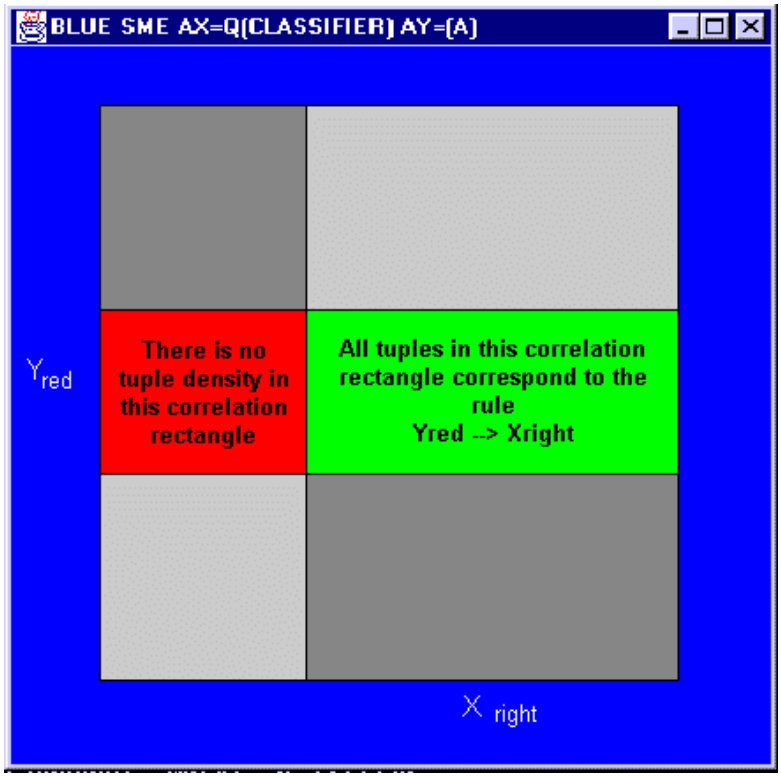
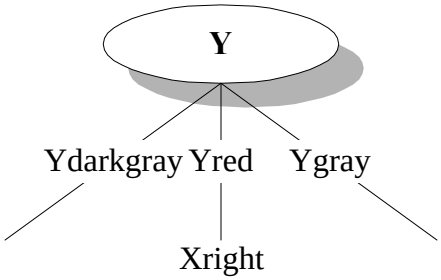


Figure 6: Root Node of Decision Tree



$Y_{red} \rightarrow X_{right}$

Figure 7: Decision Tree Root Node

Although users will detect fully correlated red/green correlation rectangle pairs as just shown, users will also experience red/grayscale correlation rectangle pairs – these represent partial correlation. This situation is represented in Figure 8. It can be seen that the classification attribute contains three classes and that the *y*-axis attribute contains two categories. The two red correlation rectangles in the top row have zero tuple density. It follows that all tuples that have the *y*-axis category represented in the top row always classify to the first class represented by the first colmun of the *x*-axis. Consequently, a leaf node can be grown from the root of the decision tree and a corresponding classification rule extracted. The resultant decision tree and rule are represented in Figure 9.

Leaf node selection entails total accountability of all tuples in a BIDS to specific classes. Three common cases are shown in Figure 10. Figure 10a demonstrates the case where a single attribute category is correlated with two classes. One leaf node, and its corresponding classification rules, would be induced in this case. Figure 10b demonstrates the case where two attribute categories are fully correlated with two classes. Two leaf nodes, and their corresponding classification rules, would be induced. Finally, Figure 10c represents the case where a single attribute category is fully correlated with a single class. As a result, a leaf node, and its corresponding classification rule, would be induced.

Multitudinal cases are experienced when combinations of BIDS must be compared and contrasted. This situation occurs with the selection of internal nodes of a decision tree or during the selection of initial splitting attribute of a decision tree. Figure

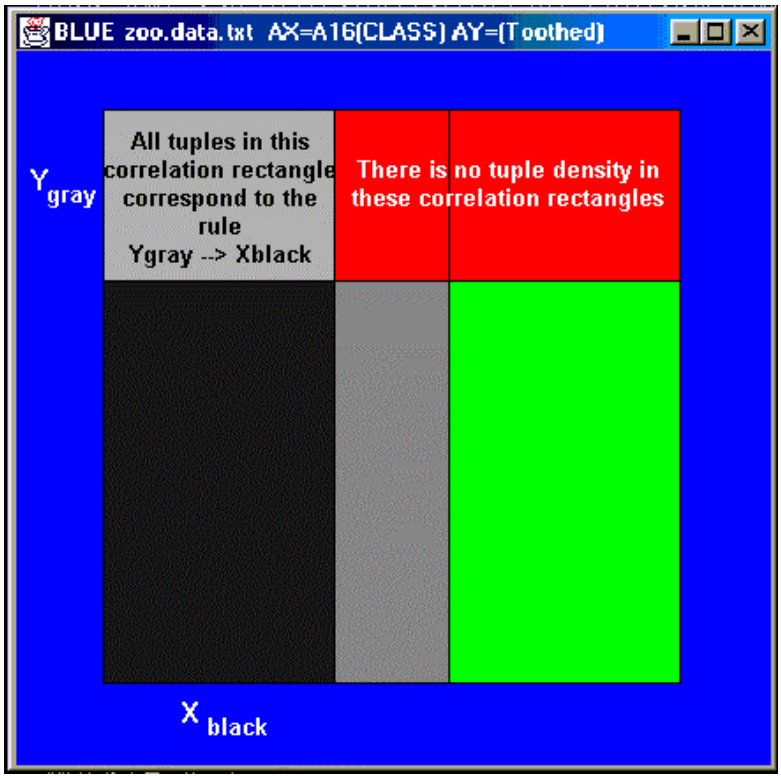
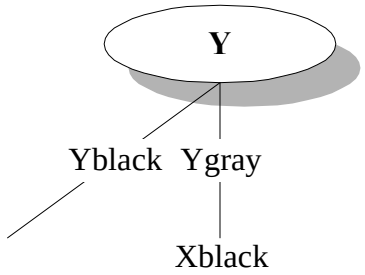
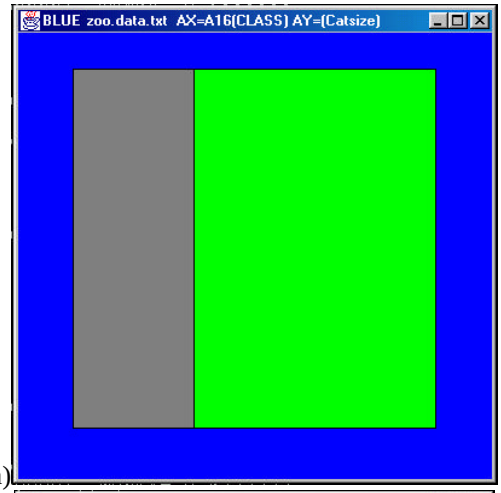


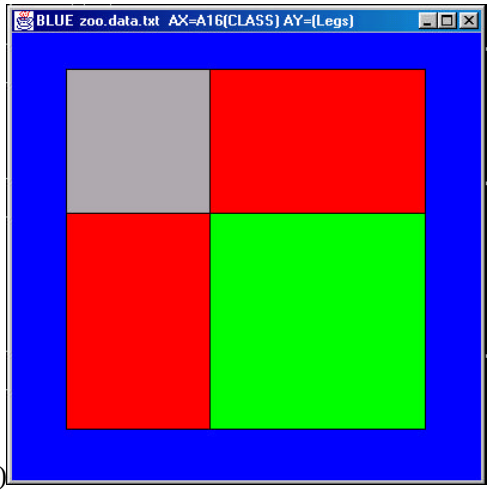
Figure 8: Tuple Density Example



$Y_{gray} \rightarrow X_{black}$

Figure 9: Determination of Leaf Xblack

a)


b)


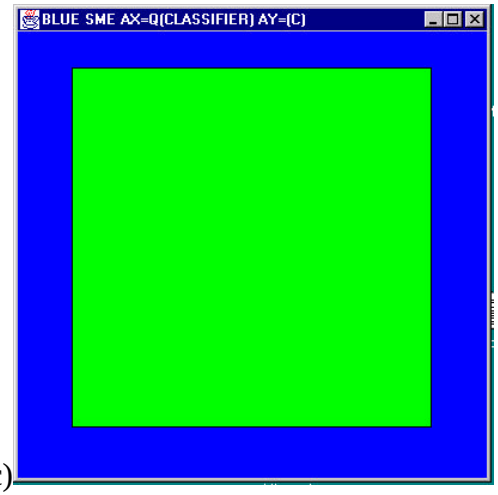
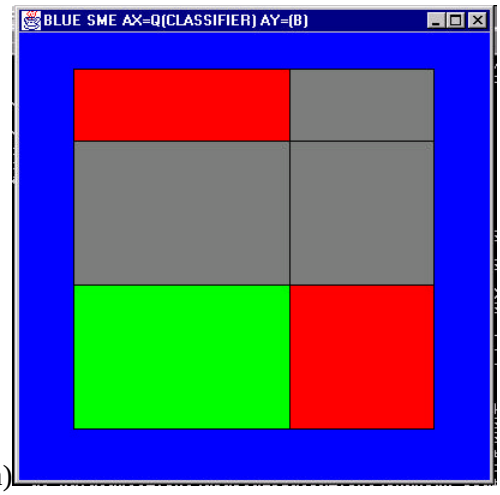
c)


Figure 10: Singular Tuple Examples

11 is used to demonstrate the decision-making process of selecting BIDS in the multitudinal case. Figure 11a is a BIDS displaying attribute A_1 and the classification attribute and Figure 11b represents a BIDS of attribute A_2 and the classification attribute. The goal is to determine which attribute should be split upon for growing the next node in the decision tree.

If Figure 11b were the only BIDS being analyzed, it would simply be a matter of following the processes outlined in the singular cases above to extract leaf node(s) and their corresponding classification rule(s). The challenge associated with situations that involve multiple BIDS is that the selection of a fully correlated BIDS in preference to another BIDS, that is only partially correlated, could result in masked attribute interaction further down the decision tree. Consequently, key relationships may not be discovered. In the current example, because Figure 11b is a fully correlated BIDS and Figure 11a is not, Figure 11a is selected as the next splitting attribute. This results in two leaf nodes and two classification rules being extracted.

Figure 12 shows a more complicated multitudinal example. In this case there is a comparison of 15 predictor attributes from the zoo database (found in appendix A). The goal is to determine which attribute should be selected as the initial splitting attribute. For purposes of clarity, a BIDS that displays the *fins* and the classification attribute will simply be referred to as the *fins* BIDS, a BIDS that displays the *legs* and the classification attribute will simply be referred to as *legs*, and so on. BIDS *breathes* and *fins* are immediately eliminated from consideration because fully correlated BIDS, when considered for internal nodes, could possibly result in missed attribute interaction of the remaining 14 attributes. Attributes that contain small red correlation rectangles are also

a) 

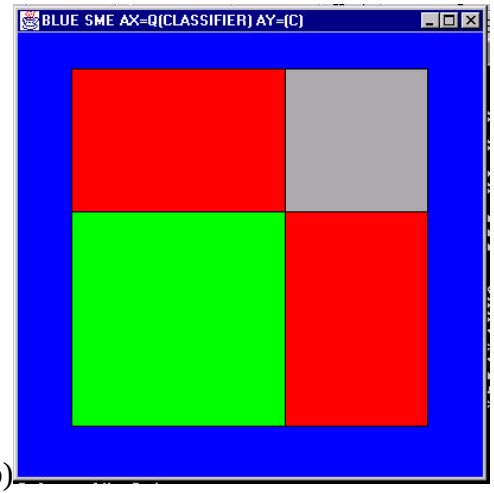
b) 

Figure 11: Multitudinal Tuple Example

eliminated because they do not represent good delineation of attribute categories and corresponding classes. This removes *eggs*, *milk*, *aquatic*, *predator*, *toothed*, *venomous*, *domestic*, and *catsize* from consideration.

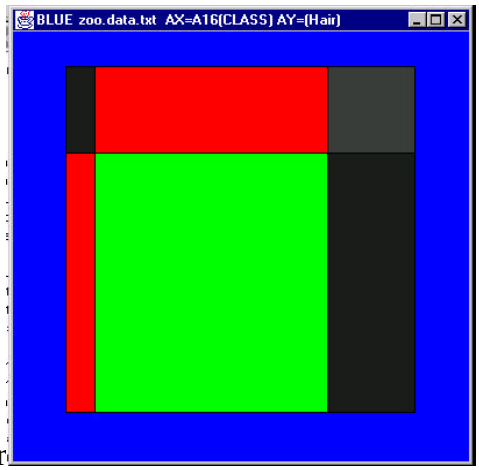
Consequently, the only remaining BIDS that are interesting are *hair*, *feathers*, *backbone*, *legs*, and *tail*. *hair*, *backbone*, *legs*, and *tail* possess similar correlation rectangles. As a result, *feathers* is selected as the next splitting attribute – it possesses the best delineation of correlation rectangles in terms of magnitude along both the *x*-axis and *y*-axis. At this point the BIDS would be analyzed with the singular guidelines outlined above to see if any node(s) and rules(s) could be extracted.

The utilization of singular and multitudinal BIDS facilitates the induction of decision tree nodes and the extraction of corresponding classification rules. The analysis and application of this process can be expressed in algorithmic terms. The algorithm that describes decision tree node(s) and classification rule(s) extraction for the singular cases is summarized in Figure 13. The algorithm that describes decision tree node(s) and classification rule(s) extraction for the multitudinal cases is summarized in Figure 14.

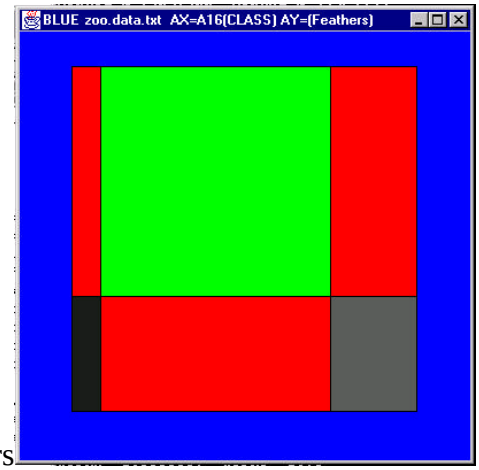
Step 3 – Software Prototype

The software prototype created in this dissertation utilized software engineering's iterative evolutionary prototyping model. This approach was selected because it allows for the construction of prototypes based upon known requirements and an understanding of the problem domain. As BIDS were analyzed and decision trees constructed, the prototype was refined and evolved.

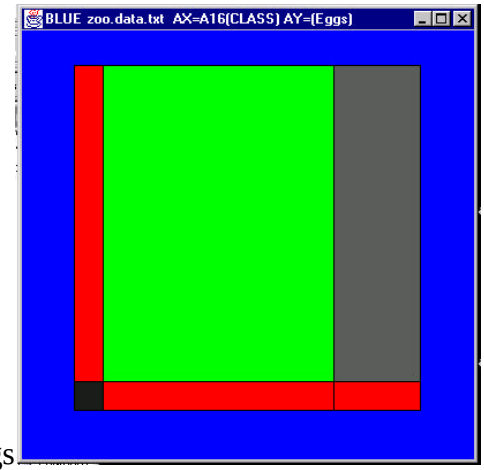
Java was used as the programming language of choice. A number of factors led



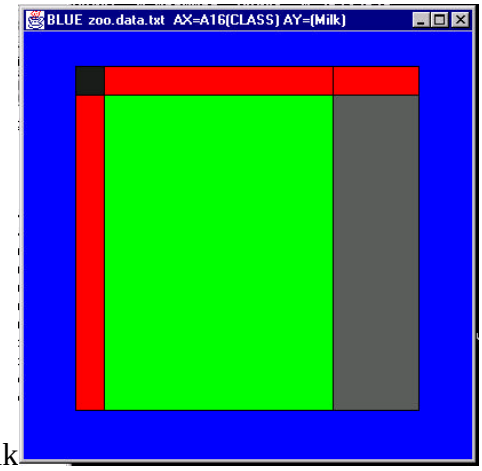
hair



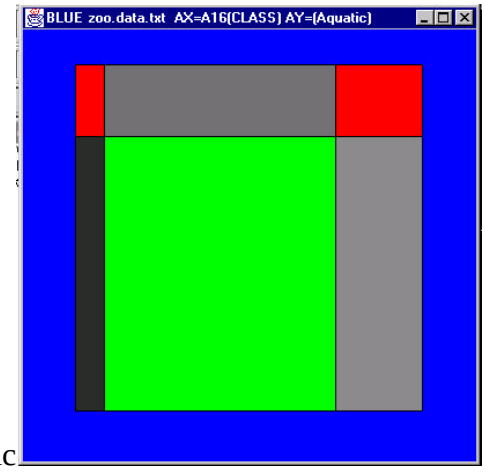
feathers



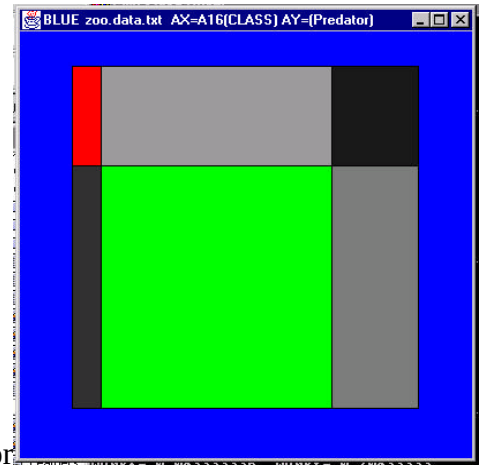
eggs



milk

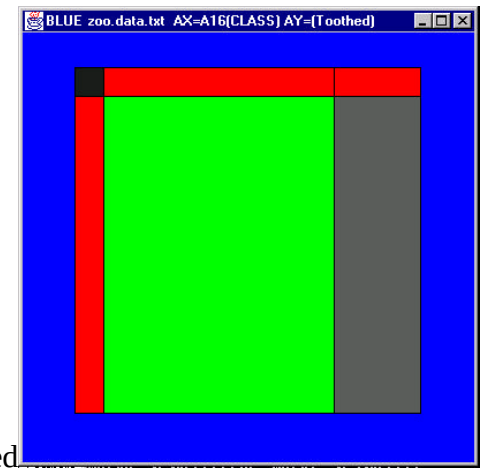


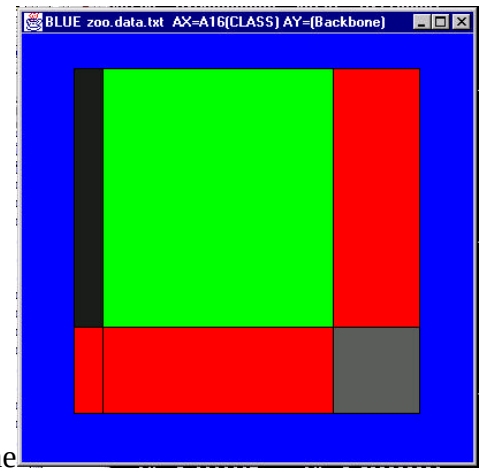
aquatic

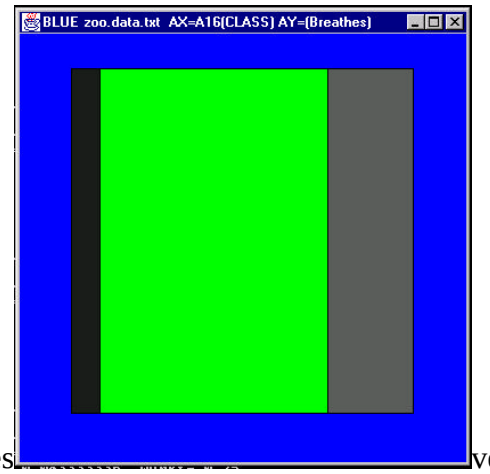


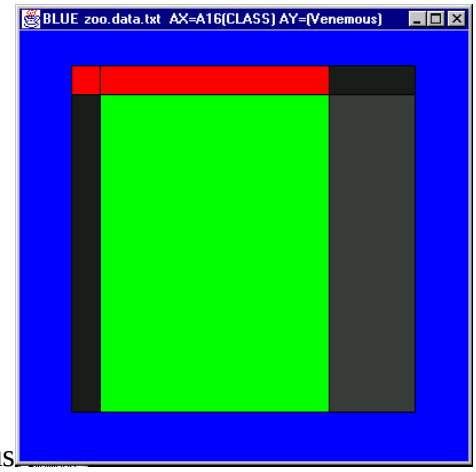
predator

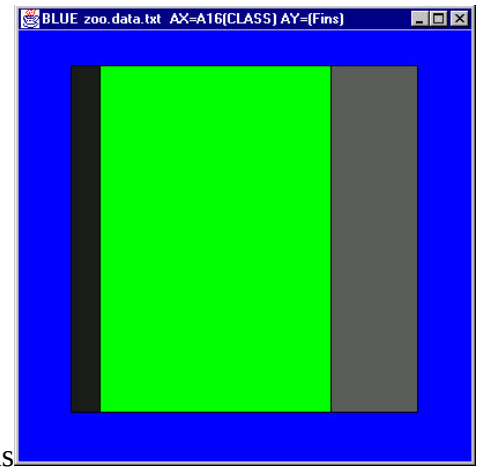
Figure 12: Multitudinal Tuple Example, 15 Predictor Attributes

toothed 

backbone 

breathes 

venemous 

fins 

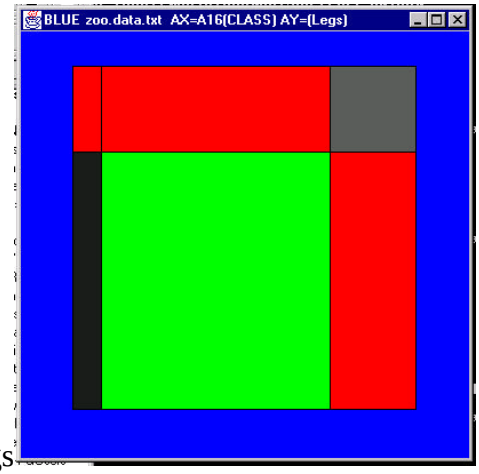
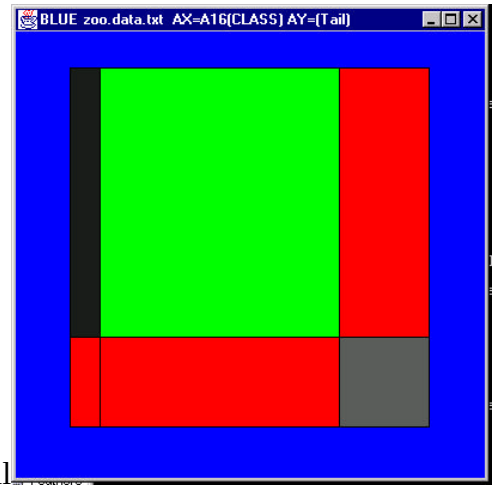
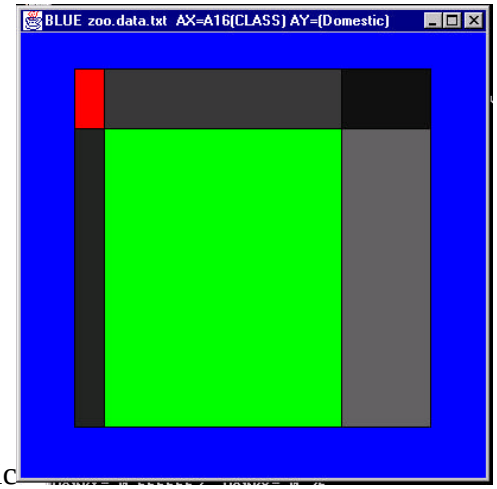
legs 

Figure 12: Multitudinal Tuple Example, 15 Predictor Attributes (continued)

tail 

domestic 

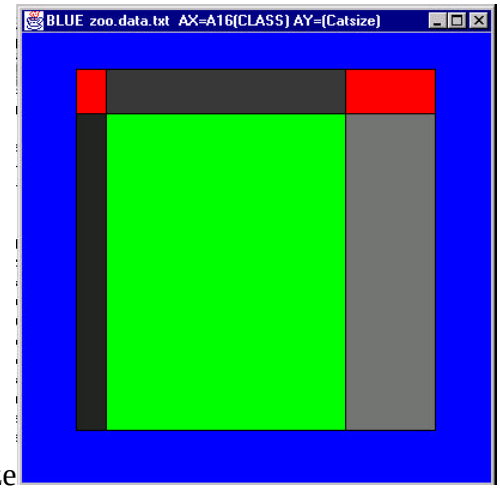
catsize 

Figure 12: Multitudinal Tuple Example, 15 Predictor Attributes (continued)

Algorithm: (RSB) - Reading Singular BIDS
Input:
i) set of singular BIDS
Output:
i) Extracted decision tree node(s)
ii) Extracted classification rule(s)
Method:
Step1: *Select task-relevant data by looking at set of singular BIDS.*
Step2: *Call procedure singular_BIDS.*
Procedure: *singular_BIDS*
BEGIN
IF BIDS are fully correlated
THEN extract decision tree node(s) and corresponding classification rule(s).
ELSE BIDS are not fully correlated
IF any row or column of a BIDS is fully correlated
THEN extract decision tree node(s) and corresponding classification rule(s).
END {*singular_BIDS*}

Figure 13: Reading Singular BIDS (RSB) Algorithm

Algorithm: (RMB) - Reading Multitudinal BIDS
Input:
i) set of multitudinal BIDS
Output:
i) Extracted decision tree node(s)
ii) Extracted classification rule(s)
Method:
Step1: *Select task-relevant data by looking at set of multitudinal BIDS.*
Step2: *Call procedure multitudinal_BIDS.*
Procedure: *multitudinal_BIDS*
BEGIN
IF BIDS all have the same cardinality
THEN select BIDS that are fully correlated
ELSE BIDS do not have the same cardinality
Select BIDS of highest cardinality that are fully correlated
Select single BIDS with maximum delineation in the *x* and *y*
Dimensions.
Call *singular_BIDS*
END {*multitudinal_BIDS*}

Figure 14: Reading Multitudinal BIDS (RMB) Algorithm

to this selection. First, Java's object-oriented programming paradigm naturally facilitates the construction of a data visualization system consisting of many interacting components. Next, Java lends itself well to applications where CPU performance levels are not the primary concern. The application developed in this paper did not require real-time response rates – the interpreted nature of the language offerered sufficient performance levels. Also, Java's platform independence allows for portability, should the software require additional development. Further, Java is an Internet-friendly language, and like the portability issue, if the system were to be advanced beyond the prototype created in this dissertation, a distributed version of the program could be developed. Finally, Java has features that simplify many tasks that would otherwise need to be addressed in another programming language such as C++. These features include convenient memory allocation, garbage collection, and access to predefined packages that allow for the quick development of Graphical User Interfaces (GUIs).

BLUE utilized a command-line interface for input. Output was represented in the form of BIDS, a decision tree, and classification rules. The decision tree graphical representation was constructed by hand. This allowed the user to explore a given data set step-by-step.

Step 4 – The Visual Data Mining Algorithm: BLUE

BLUE was constructed with a number of key requirements in mind. These included:

1) Users must possess the ability to select initial and subsequent splitting attributes.

2) A procedure for decision tree creation.

3) A procedure for extracting classification rules.

4) An ability to backtrack.

5) Guidelines for reading both singular and multitudinal BIDS.

6) The flexibility to allow the user to select any attribute splitting strategy.

7) An approach for combining requirements.

Given these requirements BLUE was created based upon a defined procedure. Below is a synopsis outlining this procedure. This is followed by a formal presentation of BLUE.

1) An initial splitting attribute was selected – based upon one of three splitting criterions; gainratio, cardinality, or personal preference.

2) The number of tuples corresponding to each class in the classification attribute were determined for the attribute under question.

3) Correlation rectangles were constructed based upon the correct normalization of tuple counts and respective categorical ownership.

4) A tuple count resident in each correlation rectangle was determined.

5) The correlation rectangle with the maximum number of tuples when compared to all other correlation rectangles was assigned the color green. Correlation rectangles possessing no tuples were assigned the color red. A grayscale linear interpolation was performed for the remaining correlation rectangles containing greater than zero tuples and less than the maximum tuples represented in the green correlation rectangle.

6) The resulting BIDS were analyzed for singular or multitudinal relationships.

7) Decision tree node(s) were induced and represented in the corresponding decision tree.

8) Classification rule(s) corresponding to the extracted decision tree node(s) were extracted.

9) The tuples representing the currently induced decision tree node were removed from the data set. If the attribute was fully represented, it too was removed from the data set.

10) Based upon the chosen splitting criteria, the next attribute(s) were selected for analysis.

11) The process began again at step two until no further tuples or attributes remained in the data set.

12) The classification rules were presented in Disjunctive Normal Form (DNF).

The Selection of Initial Splitting Attribute

BLUE provides a flexible framework from which the user can guide the selection of splitting attributes. The selection of a decision tree's root node directly influences the derivation of the remainder of the decision tree and its corresponding classification rules. Fayyad and Irani (1992) hypothesize that this is the most important aspect in top-down decision tree induction. Chapter 2 highlighted a number of major splitting strategies and their associated strengths and weaknesses.

The data sets selected for testing in this dissertation underwent three splitting strategies for the primary splitting attribute. These included gainratio, cardinality, and personal preference. Details about the resultant calculations for each splitting strategy are presented in the appendices. A synopsis of the results is also presented in Table 4.

The ability for users to select different splitting strategies in one data visualization framework highlights a significant difference among BLUE, ID3, C4.5, and CART – namely, the flexibility to guide the induction process. The aforementioned splitting algorithms have set splitting strategies. Variance, or exploration, from these strategies is restricted due to their information-theoretic or statistical foundations. BLUE, however, can support any splitting strategy because the user's discretion guides the induction process. If desired, this could include multiple splitting-strategies within the derivation of a single decision tree.

Decision Tree Induction

Chapter 2 reviewed decision tree origins, procedures for induction, and evaluation criteria. BLUE's use of decision trees for global model representation, in combination with BIDS's ability to provide local detail between attribute pairs, contributes to the advancement of the field of knowledge representation in several ways. This section focuses on BLUE's flexible approach in inducing meaningful decision tree models. These models are based upon interactive induction of decision tree nodes as well as flexible node-splitting, backtracking, and pruning strategies.

Splitting Strategy

BLUE supports both multi-way and binary (two-way) splits. Multi-way splits allow nodes to be split along all categories of a given attribute. C4.5 supports multi-way splits. This results in decision trees that are not as deep as those produced with binary splits. Binary splits produce decision trees that are represented by a series of expressions that evaluate to true or false answers. This methodology results in parent nodes possessing two child nodes. Consequently, these trees are typically simple to comprehend but tend to be large when compared with those produced by multi-way splits. CART supports binary splits.

BLUE is not limited to multi-way or binary splits. Depending upon the composition of an attribute's categories, both approaches are supported. This allows for decision trees to be constructed that are combinations of the two splitting strategies and subsequently yield more meaningful decision tree models.

Backtracking

BLUE supports backtracking based upon visualization of the induced decision tree. If a researcher wishes to explore the interplay between attributes, he or she may backtrack to a specific node and select a different induction path to see how the resultant model is affected. The decision to backtrack at a given node is based upon the researcher's analysis of the induced decision tree and the validity of the corresponding classification rule(s) leading up to that point. Hence, BLUE intuitively and interactively facilitates a form of top-down induction with backtracking guided by the user. This

flexibility is not possible in ID3, C4.5, or CART because one cannot retrace and redirect the splitting nodes.

Pruning

Traditional decision tree induction algorithms (i.e. CART and C4.5) are built in two stages: a growth stage and a pruning stage. The growth stage is more computationally costly than the pruning stage because it requires multiple passes through a given database table. BLUE grows complete, overfitted, decision trees and does not provide an explicit pruning procedure. This approach allows users to select a preferred pruning strategy given a fully grown decision tree. In this respect, BLUE is similar to traditional decision tree algorithms that utilize separate growth and pruning stages. However, depending on the needs and focus of the researcher, BLUE's flexible induction process may arguably result in enhanced decision tree models (see the section in Step 2 above entitled *BLUE's Independence Diagrams* for a discussion of BIDS as well as the section above, *Backtracking*).

Each node of a decision tree is interactively induced by providing a representation of what that portion of the tree will look like when the tree is fully realized. At each step of the induction process only two attributes are examined – not all combinations of attributes. As nodes are selected for splitting, only those tuples that contribute to the classification attribute's classes are considered. Consequently, BLUE is based upon a growing phase that analyzes the complete set of tuples only at the time the initial splitting attribute is chosen. After the decision tree has been fully grown, the user may select whichever pruning methodology he or she prefers.

Classification Rules – Motivation, Format, and Presentation

Data visualization systems that attempt to display association rules via graphical means are limited by their format representation (Iizuka et al., 1998; Liu et al., 1998; Yoda, Fukuda, Morimoto, Morishita, & Tokuyama, 1997). Classification rules were selected as an alternative form of concept representation because they can be easily extracted from decision trees in the form of an implication. It was also desired that the minimized set of rules be relevant to the problem domain – decision trees provide a global context upon which the rule can be understood.

It is well documented that a comprehensive set of association rules can be induced from a data set utilizing various algorithms. The problem that arises from such a solution is as the number of rules induced increases comprehensibility is adversely affected (Klemettinen et al., 1994; Liu et al., 1998; Piatetsky-Shapiro & Frawley, 1991b). In addition, numerous rules tend to duplicate one another and some may provide conflicting information – the resulting rule-set requires a domain-expert to intervene. In an effort to minimize the problems associated with this costly analysis, measures of support and confidence were applied in an attempt to induce only the "most interesting" rules – or those most preferred by the users (Agrawal & Srikant, 1995). Solely relying on support and confidence measures to derive rule-sets can lead to significant problems.

Wang, Zhou, and He (2000) point out that pruning measures based upon support to reduce the number of rules to those that are the "most interesting" suffer from a specific problem; namely, rules that possess high support tend to have low confidence. In addition, user-defined support threshold values can be difficult to select. In many cases

reasonable support levels are unknown in advance without performing empirical tests of the data set.

Another problem with association rules involves relationship frequency occurrence. Cohen et al. (2000) point out that rule induction algorithms such as *apriori* are only effective in applications where relationships occur frequently, market-basket analysis for instance. Even though *apriori* and other algorithms rely upon a measure of support, some relationships that have low support but high-confidence are culled – for some applications – these are the "most interesting" rules. Application domains the authors view this as problematic include data mining, web mining, clustering, and collaborative filtering.

Consequently, induction methods such as C4.5 and CART tend to be application-specific, automated, and in some cases, result in the "most interesting" rules being culled. In effect, the induction of association rules in this setting is blindly directed. An interactive data visualization method that allows users to guide the development of the model could be useful.

BLUE may be helpful in rule induction by providing a framework from which a subset of the association rule problem can be addressed. This subset of rules is induced through classification mining (Wang et al., 2000), classification rule mining (Liu et al., 1998), and classification rule-discovery (Freitas, 2000). Classification rules have the following form:

Definition 3.8 For a given relation M that possesses non-class attributes A_1, A_2,…, A_m and class attribute C, a case can be represented by the vector $<a_1, a_2,\ldots, a_m, c_j>$ where a_i

is a value of A_m and c_j is a class from C. A rule can then be expressed with $A_{i1} = a_{i1} \wedge \ldots \wedge A_{ik} = a_{ik} \rightarrow c_j$ where each attribute cannot occur more than once.

BLUE extracts classification rules as the decision tree is induced. The section above, *BLUE's Independence Diagrams (BIDS)*, describes the process utilized to extract decision tree node(s) and corresponding classification rule(s). This process continues until the decision tree is fully grown. At this point the user may apply a pruning technique to cull the decision tree and rules – for instance, the measures of support and confidence. When pruning has stopped, the resultant decision tree is presented and the classification rules are presented in Disjunctive Normal Form (DNF)

BLUE's Combined Approach

BLUE's individual approaches for reading BIDS, growing decision trees, and extracting classification rules have been outlined in prior sections. BLUE's benefits are magnified when these approaches are considered together.

BLUE's decision tree emphasis indicated that only limited passes through the data set were required to induce a given tree. The only time a complete pass through the data set was needed, after the root node was induced and its children determined, would be if backtracking were to occur.

Another benefit is the fact that domain and non-domain experts alike could use BLUE. Guidelines were established for novice users so that adequate models could be induced without the benefit of expert knowledge. Consequently, the researcher would be provided with a tool to induce a combined rule/tree-based model (i.e. go from general to

specific form), possibly entailing unknown relationships prior to the induction process. Finally, by providing a method for viewing local detail within a global context, the issue of comprehensibility was addressed.

BLUE: The Complete Algorithm

This section provides the complete visual data mining algorithm called BLUE. The algorithm, presented in Figure 15, outlines the steps necessary to apply BLUE to relational databases and tables.

Illustrative Example

The following is an example to illustrate how BLUE is employed to construct a decision tree and its corresponding classification rules based upon the data in Table 2[12].

The input for BLUE is as follows:

1) Relational database table: Table 2.

2) Specification of classification attribute: *PLAY*.

3) Splitting strategy: *Personal Preference.*

The execution of BLUE is as follows:

Step 1: The task-relevant data is displayed in a single relational table (Table 2).

Step 2: Call procedure *BLUE*.

i) The initial splitting attribute is chosen to be *OUTLOOK*.

[12] The data set used in this example was adapted from Quinlan (1986) and presented in Wu (1995).

Algorithm: BLUE

Input:

i) relational database or single relational table
ii) specification of classification attribute
iii) splitting strategy

Output:

i) Decision tree
ii) Classification rules
iii) BLUE's Independence Diagrams (BIDS)

Method:

Step1: *Select task-relevant data by relational query or by presenting a single relational table to BLUE.*

Step2: *Call procedure BLUE.*

Procedure: *BLUE*

```
// Suppose the table from Step1 consists of a set of attributes A_i where
// 2 ≤ i ≤ n and n is the number of attributes in the set.  T represents the
// number of tuples in the set, c represents the number of classes in
// classification attribute C.
BEGIN
    Determine initial splitting attribute
    WHILE(T > 0) DO
    BEGIN
        FOR each attribute to be evaluated
            Calculate number of tuples per category
                corresponding to each c
            Create BIDS based upon normalization of tuple
                counts and categorical ownership
            IF BIDS is singular
                Call procedure singular_BIDS
            ELSE BIDS is multitudinal
                Call procedure multitudinal_BIDS
        Grow decision tree node(s)
        Remove tuples corresponding to induced node(s) and
            attribute(s) if appropriate
        IF there are no more attributes to be evaluated
        THEN exit loop
        ELSE select attributes to be evaluated next iteration and
            loop again
        END{FOR}
    END{BEGIN}
END {BLUE}
```

Step3 *Transform the final relation represented with classification rules into Disjunctive Normal Form (DNF)*

Figure 15: BLUE – Algorithm

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
Sunny	Hot	High	False	Don't Play
Sunny	Hot	High	True	Don't Play
Overcast	Hot	High	False	Play
Rain	Mild	High	False	Play
Rain	Cool	Normal	False	Play
Rain	Cool	Normal	True	Don't Play
Overcast	Cool	Normal	True	Play
Sunny	Mild	High	False	Don't Play
Sunny	Cool	Normal	False	Play
Rain	Mild	Normal	False	Play
Sunny	Mild	Normal	True	Play
Overcast	Mild	High	True	Play
Overcast	Hot	Normal	False	Play
Rain	Mild	High	True	Don’t Play

Table 2: Saturday Morning Relation

ii) A calculation is performed to determine how many tuples reside within each category of *OUTLOOK* that corresponds to each class of the classification attribute. In this case, *Sunny* has two tuples corresponding to *Play* and three tuples corresponding to *Don't Play*. *Overcast* has four tuples all corresponding to *Play*. Rain has three tuples corresponding to *Play* and two tuples corresponding to *Don't Play*.

iii) A BIDS is created and displayed in Figure 16 based upon the aforementioned tuple distribution. The *x*-axis represents the classification attribute *PLAY* while the *y*-axis represents the attribute *OUTLOOK*.

iv) In this case the BIDS is singular so the procedure *singular_BIDS* is called to extract the decision tree node(s) and corresponding classification rule(s). It can be seen that *Overcast*, the category in the center horizontal direction of the BIDS is fully correlated. Consequently, Figure 17 shows the extracted node and classification rule.

v) The tuples corresponding to the *Overcast* correlation are removed from the relation. Attribute *Overcast* cannot be removed from the relation because tuples corresponding to *Sunny* and *Rainy* remain. Table 3 shows the resultant relation.

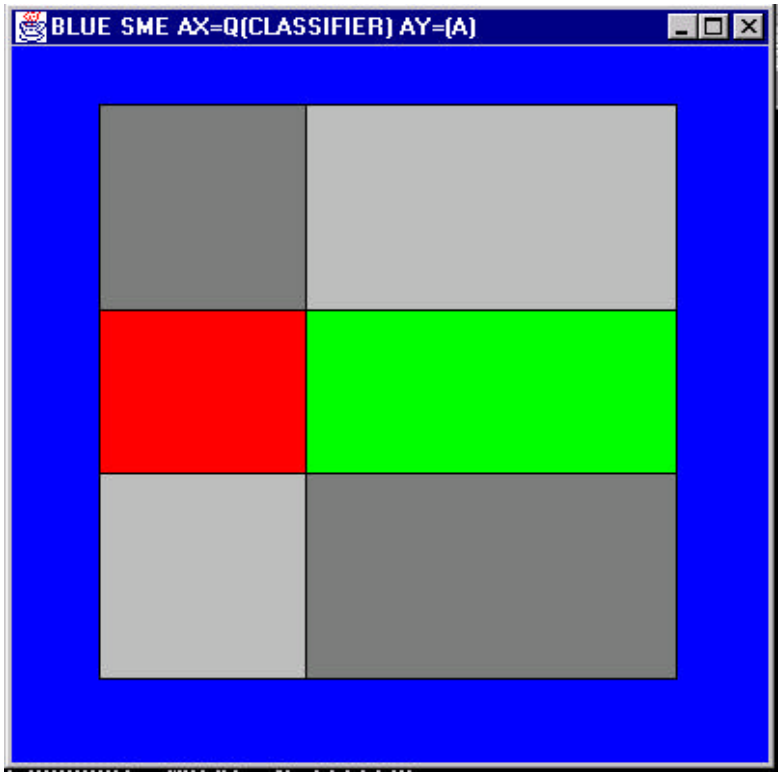
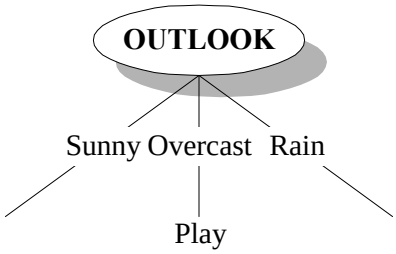


Figure 16: *OUTLOOK* and *PLAY* Attributes



Overcast→Play

Figure 17: Induced Decision Tree Node and Classification Rule

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
Rain	Mild	High	False	Play
Rain	Cool	Normal	False	Play
Rain	Cool	Normal	True	Don't Play
Rain	Mild	Normal	False	Play
Rain	Mild	High	True	Don't Play
Sunny	Hot	High	False	Don't Play
Sunny	Hot	High	True	Don't Play
Sunny	Mild	High	False	Don't Play
Sunny	Cool	Normal	False	Play
Sunny	Mild	Normal	True	Play

Table 3: *OUTLOOK*'s *Overcast* Tuples Removed

vi) Since non-classification attributes and tuples remain in the relation the process must be repeated again.

vii) Select the remaining non-classification attributes for comparison in the next iteration until there are no more tuples in the relation.

viii) The final step is to represent the resultant classification rules in Disjunctive Normal Form (DNF). Figure 18 shows the final decision tree and relation.

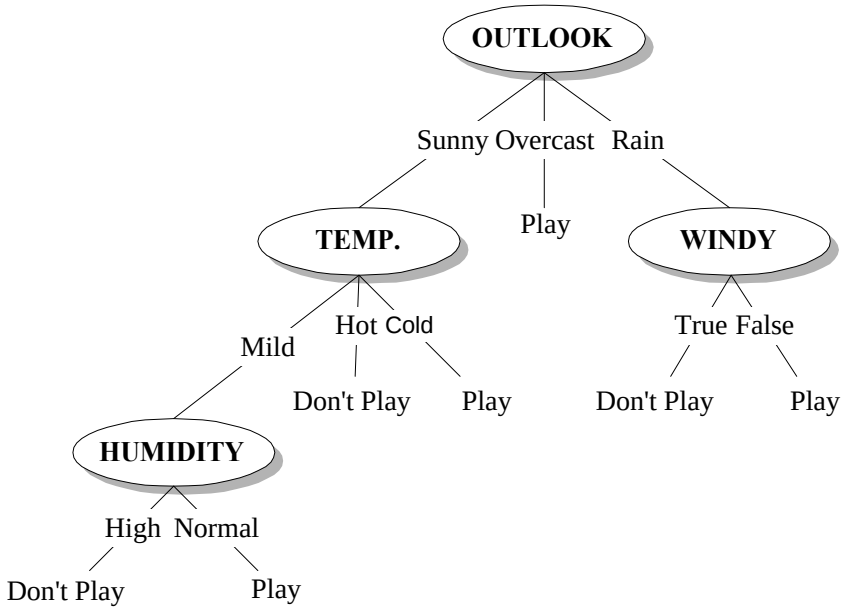
Step 5 – Formats for Presenting Results

Table 4 (in chapter 4) lists the summarized results of the empirical testing that was carried out in this dissertation. Specific results pertaining to entropy, information gain, cardinality, and personal preference calculations are presented in Appendices A-E.

Step 6 – BLUE's Reliability and Validity

BLUE is a visual data mining algorithm that represents one potential direction the data mining field might take. Although this study cannot be generalized beyond the scope of the data sets tested, BLUE was used to detect patterns and regularities in data sets and build representational models for such findings. Consequently, BLUE's reliability should be consistent between test runs for the data sets selected in this dissertation.

BLUE's validity is based upon the number of leaf nodes/classification rules created and its comprehensibility. It was discerned whether or not BLUE classified cases correctly based upon the cases presented in the data set. These results would be expected



(Overcast→Play) ∨
(Rain ∧ Not Windy→Play) ∨
(Rain ∧ Windy→Don't Play) ∨
(Sunny ∧ Hot→Don't Play) ∨
(Sunny ∧ Cold→Play) ∨
(Sunny ∧ Mild ∧ Humid→Don't Play) ∨
(Sunny ∧ Mild ∧ Normal Humidity→Play)

Figure 18: Final Decision Tree and Relation in DNF

because BLUE iterates from the global view of all tuples contained in the data set including the last tuple in the data set.

Step 7 – Selected Data Sets and Experiment Guidelines

The focus of this dissertation was on extracting global models of relational database tables based upon reading BIDS. The building of these models required selecting representational data sets for experimental purposes. In the event that a chosen data set was characterized by numeric values, attribute value ranges were defined and tuples were assigned accordingly – a process referred to as binning.

The data sets utilized in this study were obtained from the University of California at Irvine's (UCI) Machine-Learning Repository (MLR). These data sets have been used in numerous empirically-based experiments (In fact, UCI's MLR has a history as being the *de facto* source of data sets for the machine-learning field). These carefully selected data sets enabled a comparison of BLUE with other research based on the same data sets created with C4.5. Four data sets were selected for comparison, including Zoo, Iris, Glass, and Voting (see the appendices for details regarding each data set).

A well-known example from the literature was selected for illustrative purposes[13] and appears in the section above, *Illustrative Example.* Since each of the four data sets were relatively small in size in that none contained more than 435 tuples, it was deemed useful to utilize an additional data set. The data set selected for this evaluation was the Titanic data set (also from UCI's MLR). This data set contained 2201 tuples encompassing four attributes.

[13]This example originally appeared in Quinlan (1986).

Two experiments were conducted. The first experiment involved the application of BLUE, as defined in figure 15, to each of the aforementioned data sets, three times. Each time a data set was tested, the calculation of the initial splitting attribute was changed. The three strategies included gainratio, cardinality, and personal preference. This resulted in fully grown, overfitted, decision tree models (these models are presented in the appendices). The goal was to determine if BLUE was a viable alternative to other approaches of inducing comprehensible decision trees.

The second experiment reduced the resultant rule-sets from experiment one. This required implementing a simple pruning strategy. This strategy measured the support for each classification rule derived from the fully grown decision tree. The rule-set was reduced by discarding rules with smallest support until the number of rules matched the reported number in Wang et al. (2000). The goal was to determine if there was a correlation between support levels among the three splitting techniques.

Further details about each data set, experimental approaches, and the results are presented in the appendices. Summarized results are presented in Table 4.

Resources

Modest computing resources were necessary to facilitate the completion of the experiments in this dissertation. All development and testing was completed on a single 600 MHz PIII running Windows 98. Java 1.3.0 was utilized along with Sun Microsystems Forte for Java Community Edition 1.0 for the development environment. Nova's distance library services were utilized to help obtain relevant journals, books, and

periodicals. Regular communication with the dissertation committee helped direct the research and facilitate advancement in the field.

Summary

A well-constructed visual data mining algorithm entails many requirements. In terms of functionality, users should possess the ability to interactively explore a data set so that previously unknown relationships might be detected and/or predictive models constructed. Consequently, total reliance on fully-automated algorithms do not fulfill visual data mining's mission. As a result, a data visualization methodology that provides local detail from within a global context of the data set is desirable. This methodology should also contain a facility to revisit induced nodes to facilitate maximum understanding of the data set.

This chapter developed a visual data mining algorithm called BLUE from which the aforementioned goals could be fulfilled. A type of independence diagram called a BIDS was defined and guidelines for reading images utilized with them were articulated. BIDS differ from independence diagrams in that they utilize color, a different tuple density measure, and are focused solely on categorical attributes, or discretized continuous attributes. BIDS are used as the visualization method to provide local detail to the user.

Decision trees were used as the visualization method to provide global context to the user – due to their ease of comprehension, their ease of programmability in a computer, and their generalization ability. Consequently, the final graphical model

representing the knowledge contained in a relational database table is presented in decision tree form.

The splitting strategy utilized in the induction of any decision tree is important. It was deduced that the most difficult splitting node to select is the first one, the root of the tree (Fayyad & Irani, 1992; Quinlan, 1986; Ankerst et al., 1996; Breiman et al., 1984; Lewis, 1962). It was decided to utilize three different measures to select the initial splitting node in the experiments carried out in this dissertation. These methods included gainratio, cardinality, and personal preference.

Gainratio is an information-theoretic approach that attempts to maximize information gain and minimize uncertainty between attributes. Cardinality takes advantage of the intrinsic cardinalities of attributes by selecting attributes with the smallest cardinality. Personal preference is a more subjective approach than gainratio or cardinality because it relies on user selectivity preferences. For example, an expert user might understand certain relationships between attributes in a given data set. In this case, the user might want to purposefully direct decision tree growth in an effort to discover new relationships in the data. Another example of the personal preference approach's subjectivity can be exemplified by a non-expert user, who may wish to select a certain attribute as the initial splitting node (since the user possesses little, if any, knowledge about potential relationships between attribute pairs). The flexibility in being able to select the initial splitting attribute allows the user to rely upon a model based upon his or her research interests.

It was also determined that it would be useful to induce a subset of association rules called classification rules as a decision tree was created. Classification rules

facilitate the extraction of rules that naturally follow the representation of the knowledge in the decision tree; thus facilitating the extraction of rules that are interesting. As the decision tree was grown, classification rules were extracted. The resultant rule set was rank-ordered and a reduced set of classification rules was produced through utilization of a simple pruning measure – this procedure is exhibited in experiment 2 of this dissertation.

The methodology outlined in this chapter was empirically tested on five data sets. These data sets were tested three times, inline with the initial splitting attribute strategy mentioned above. Another data set, the Saturday morning data set, was processed and presented in the section above entitled *Illustrative Example*.

A number of other items were articulated in this chapter. These included guidelines for reading BIDS, resource requirements, a test methodology for empirical testing, a composite definition of BLUE in algorithmic terms, and BLUE's reliability and validity.

Chapter 4

Results

Introduction

The purpose of this dissertation was to develop an interactive visual data mining algorithm. The overriding factor for the selection of this topic was the desire to address the problem of exponential growth in the number of, and size of, databases. This growth has resulted in many organizations lacking the resources and time necessary to analyze and make decisions based upon known and previously unknown factors that might be in their data. Consequently, efforts to develop an algorithm that would interactively facilitate the construction of global models by viewing local detail between attribute pairs in relational databases ensued.

Chapter 3 outlined the methodologies utilized for the development of BLUE. This included definitions for BLUE's operational environment, guidelines for reading BLUE's Independence Diagrams (BIDS), a procedure for the production of both decision trees and classification rules from BIDS. In addition, procedures for empirically evaluating BLUE with five data sets were presented. Two experiments were carried out with BLUE – the results and analysis are presented in this chapter.

Analysis

Background Information

BLUE was developed as an interactive visual data mining algorithm. It differs from other information-theoretic or statistical decision tree inductive methods in a number of ways. First, BLUE is not fully-automated, allowing the user, domain expert or non-domain expert, the ability to backtrack and/or explore patterns and regularities in the target data set. Second, BLUE uses a visual means of displaying correlation of attribute pairs in relational data sets – the images are referred to as BIDS. In addition, a visual global context of the derived model is provided in the form of a decision tree. As the decision tree is induced classification rules are extracted. This allows for a thorough investigation of a data set to take place. Finally, BLUE was designed in a modular fashion to accommodate user-specified splitting and pruning strategies.

BLUE was applied to five data sets. These data sets were carefully selected to facilitate an empirical evaluation of BLUE. Consequently, BLUE was subjected to two experiments. The first experiment consisted of utilizing BLUE to extract three decision trees for each given data set, each based upon a different technique for the selection of the initial splitting attribute of the decision tree. Once the root of each decision tree was selected the remainder of each tree was induced using BIDS – corresponding classification rules were extracted as nodes were induced. The second experiment required the application of a simple pruning method to the classification rules extracted in the first experiment. This facilitated the culling of rules based upon a support level and allowed for comparison with C4.5. The results of these tests are compared, contrasted, and presented in this chapter.

Data Set Analysis

The data sets selected for evaluation in this dissertation consisted of Zoo (Appendix A), Iris (Appendix B), Glass (Appendix C), Voting (Appendix D), and Titanic (Appendix E). BLUE was designed to work with relational data sets that contain primarily categorical attributes. Consequently, it was necessary to discretize all non-classification attributes in each data set that did not contain solely categorical attributes. The Iris and glass data sets both required complete discretization of all non-classifying attributes. A linear interpolation between the lowest and highest ranges was selected and assigned to specific categories (these discretized ranges are shown in Appendices B and C, respectively).

A set of three tests were performed on each data set – they differed by the method used to select the initial splitting attribute. The three techniques included the information-theoretic approach used in C4.5 called gainratio, a cardinality measure, and personal preference. The information-theoretic approach required calculations for entropy, information-gain, split-information, and gainratio for each attribute in each data set. The attribute with the largest gainratio value was selected as the initial splitting attribute. In the case of cardinality, the attribute with the smallest cardinality was selected as the initial splitting attribute. If there were a tie between multiple attribute cardinality values, a random selection among the tying attributes was made. A summary of this work is presented in tabular form in each of the Appendices.

After each initial splitting attribute was determined, BLUE's algorithm, defined in Figure 15, was applied. BIDS were used for the induction of the remaining decision trees that utilized gainratio and cardinality initial splitting node techniques. The personal

preference initial splitting node technique also utilized BIDS but, in addition, allowed user selectivity to provide for a more explorative induction of a decision tree. This facilitated use of background knowledge. Decision trees were completely induced with no pruning (these decision trees are shown in Appendices A-E). Support levels were determined for each derived classification rule. These rules were then rank ordered and culled until the number of rules matched that found in Wang et al. (2000).

Findings

An analysis of experiment 1 consisted of a comparison of the number of decision tree leaf nodes and rules extracted when BLUE was applied to three different splitting strategies. Experiment 2 utilized the data from experiment 1 to apply a pruning technique that utilized a simple support measure. This data was compared with results from Wang et al. (2000) and Frank and Witten (2000). The goal was to determine if there was a clear correlation among support levels and the three initial splitting strategies. Table 4 contains a summary of the results from experiments 1 and 2.

Experiment 1

The purpose of experiment 1 was to determine if the combination of BLUE and any one of the three attribute splitting strategies resulted in decision tree models with smaller number of leaf nodes, smaller number of classification rules, and more comprehensible models. The experiment was carried out as described in the *Data Set Analysis* section above.

Data Set	# Tuples/ # Attributes	C4.5* Leaves	C4.5* Rules	BLUE (Unpruned) Leaves/Rules	BLUE (Pruned) Leaves/Rules	BLUE Min. Pruning Support (%)
Glass	214/9	27	14.6±0.6			
GR				72	14.6	1.3
CD				55	14.6	2.3
PR				55	14.6	4.7
Avg				61±9.8		2.8±1.75
Iris	150/4	4	5.0±0.1			
GR				11	5	2.7
CD				11	5	5.3
PR				17	5	8
Avg				13±3.5		5.3±2.65
Vote	435/16	10.2	7.0±0.2			
GR				35	7	7
CD				53	7	4.8
PR				46	7	3.9
Avg				45±9.1		5.2±1.6
Zoo	101/16	17.8	9.1±0.1			
GR				17	9	4.2
CD				18	9	3
PR				14	9	4
Avg				16.3±2.1		3.7±.64
Total Average				33.8±23.1		4.3±1.21
Titanic	2201/4					
GR				24		
CD				24		
PR				22		
Avg				23±1.2		

GR = Gain-Ratio
CD = Cardinality
PR = Preference

* The number of leaves induced with C4.5 came from Wang et al. (2000). The number of rules induced with C4.5 came from Frank and Witten (2000).

Table 4: Summarized Results

The resultant decision trees were compared and contrasted both in quantitative terms and qualitative terms (the fully-grown trees can be seen in Appendices A-E). Quantitatively, gainratio produced the smallest number of rules/leaf nodes in one data set; personal preference produced the smallest number of rules/leaf nodes in two data sets; cardinality tied with gain ratio for the smallest number of rules/leaf nodes in one data set; and cardinality also tied with personal preference for the smallest number of rules/leaf nodes in one data set (see the column *BLUE (unpruned) Leaves/Rules* in Table 4). Although personal preference was slightly more successful in producing decision trees with smaller number of rules/leaf nodes, it is not reasonable to think that this is a significant finding when standard deviation is taken into account (also shown in Table 4). This result indicates that none of the splitting strategies produced decision trees and classification rule-sets that were significantly smaller than any of the other techniques.

In terms of comprehensibility, BLUE produced more comprehensible models than the gainratio or cardinality-based approaches in three out of five databases including zoo, vote, and titanic (the resultant decision trees can be viewed in Appendices A, D, and E, respectively). This conclusion is drawn because model derivation was aided due to the user's domain knowledge of each data set. Consequently, each model was created in a logical, sequential order that was familiar to the researcher.

It was not clear which initial splitting attribute produced the most comprehensible model with the other two databases, glass and iris. One explanation for the lack of comprehensibility with these data sets is that they were both comprised of discretized numeric value-ranges. These ranges may not be comprehensible by non-domain experts.

As a result, the user did not know if the selection of one category in lieu of another made sense in the derivation of each decision tree.

Experiment 2

Table 4, columns three and four, show the results Wang et al. (2000) and Frank and Witten (2000) obtained utilizing C4.5 on four of the same data sets tested with BLUE (glass, iris, vote, and zoo). The figures in Table 4 are smaller than those produced with BLUE because they represent trees that are pruned. BLUE does not contain a specific pruning methodology, however, since the fully-grown decision trees and their corresponding classification rule-sets were available from experiment 1, it was thought useful to perform an experiment based upon support levels.

The idea was to derive support levels for each set of classification rules induced in experiment 1 to see if there was a correlation between initial attribute splitting strategy and resultant support levels in classification rules induced with BLUE. First, support levels were determined for all classification rule-sets induced in experiment 1. Next, the number of rules induced with BLUE was reduced for each data set run until the same number of rules matched the number of rules derived with C4.5. At this stage, the minimum support level was identified and expressed in Table 4.

On average, a support level of 4.25% applied to classification rule-sets created with BLUE resulted in the same number of rules induced with C4.5. The gain-ratio splitting criteria required a support level of 3.8%, cardinality 3.85%, and personal preference 5.2%.

Summary of Results

The two experiments performed in this dissertation focused on different outcomes. Experiment 1 was concerned with the application of BLUE to five data sets to determine if one of the primary attribute splitting strategies would significantly affect the number of induced decision tree leaf nodes/rules as well as the comprehensibility of the derived model. Experimental results showed that use of one of the three specific splitting attribute strategies used in this dissertation did not make a statistically significant difference. Conversely, it was determined that models created with the personal preference splitting attribute technique resulted in more comprehensible models. With all parameters kept equal, this infers that models created with BLUE and a personal preference for initial splitting attribute will result in models that are easier for users to understand and utilize in practice for future prediction purposes.

Experiment 2 evaluated the classification rules derived in experiment 1. The goal was to determine if there was a correlation between support levels among the three splitting techniques with the utilization of a simple pruning measure. It was found that slightly higher support levels were needed for personal preference to induce the same number of rules as that produced with C4.5 – gainratio and cardinality required slightly smaller support levels when compared to personal preference.

Chapter 5

Conclusions, Implications, Recommendations, and Summary

Introduction

The goal of this dissertation was to develop an interactive visual data mining algorithm, compare its performance with three different primary attribute splitting strategies, and to determine if any one splitting strategy was better than the other two when realized with this system. The resulting algorithm was named BLUE. The impetus behind BLUE's creation was the exponential growth of data in the form of relational databases. Chapter 1 provided an overview of the problem domain that was to be explored. Chapter 2 reviewed the historical background, theoretical foundations, and alternative solutions that currently address the problem domain as well as articulation of what contribution this study would make to the data mining field. Chapter 3 developed the research methodology and the development framework from which BLUE was quantitatively and qualitatively developed and compared. Chapter 4 presented an analysis and the results obtained from two experiments conducted with BLUE. This chapter completes the study by drawing conclusions, describing their implications, making recommendations for further work, and summarizing the paper as a whole.

Conclusions

The results obtained in Chapter 4 verified that it is possible to combine two data visualization methods and two machine learning techniques to develop a visual data mining algorithm that both domain and non-domain users can utilize to build models of relational databases. The resultant system is a visual data mining algorithm referred to as BLUE. Models developed with BLUE were compared quantitatively and qualitatively. These results were determined experimentally.

Two experiments were performed to test the aforementioned hypothesis. The first experiment involved the induction of five data sets, each evaluated with three different splitting strategies. The goal was to determine if any one splitting strategy resulted in significantly different results from the two other approaches, and if so, by how much. The second experiment involved pruning the models that were created in the first experiment. The goal was to determine if the pruned classification rules indicated a significant difference in required support levels between primary attribute splitting techniques.

The most difficult node to induce in a decision tree is the root node (Fayyad & Irani, 1992; Quinlan, 1986; Ankerst et al., 1996; Breiman et al., 1984; Lewis, 1962). The first experiment tested the affect of deriving the root node of a decision tree with three different techniques, gainratio, cardinality, and personal preference. The measurement criterion for the test runs were number of leaf nodes/classification rules created, and comprehensibility of the resultant model.

In quantitative terms, the utilization of a personal preference primary splitting attribute indicated a slightly better outcome than that obtained with the gainratio or

cardinality approaches. Despite this result, when standard deviation was taken into account, these results did not justify the claim that personal preference produced smaller models than the other two techniques. Consequently, all three splitting strategies resulted in comparably sized decision trees and classification rules.

In terms of comprehensibility, it was found that BLUE produced more comprehensible models utilizing a personal preference initial splitting strategy than those produced with gainratio or cardinality approaches. The justification for this claim is that the models were derived utilizing knowledge of the problem domains and the ability to backtrack and reselect different inductive paths through the decision tree. Consequently, models derived utilizing personal preference were produced logically – making them easier for the user to follow and comprehend.

The second experiment was an extension of the first experiment. The goal was to determine if any one primary attribute splitting strategy required significantly more or less support when resultant decision trees were pruned with a simple support-based pruning strategy. The personal preference approach resulted in slightly higher support levels, but this could be expected as this approach produced slightly smaller decision trees. Consequently, when applied to the decision trees induced with BLUE, the initial splitting attribute technique did not produce unexpected results. As with the result outlined in first experiment, when standard deviation was taken into account a significant difference between the three initial node splitting strategies was not apparent.

BLUE was designed with many objectives in mind. These objectives resulted in the successful utilization of BLUE to perform the experiments outlined above. The design requirement stated that BLUE must be created in such a way so that it could

interactively support the simultaneous extraction of decision trees and classification rules from relational databases. This included the ability for backtracking as well as a framework where domain and non-domain users could guide the induction process. In addition, a facility was provided where attributes could be removed via personal choice so as to limit the required processing effort (see Appendices for specific instances).

The visualization methodology specified that BLUE's Independence Diagrams (BIDS) would be used to view relationships and patterns between attribute pairs. The combination of BIDS that had to be viewed during an analysis was reduced from $V_n = n(n\text{-}1)/2$ to $V_n = (n\text{-}1)$ where V_n is the number of BIDS and n is the number of attributes in a given database table (see the *Image Reduction Strategy* section in Chapter 3 for further explanation). Decision trees were utilized to provide a global context upon which the model is based. These objectives were addressed and resulted in BLUE's support of top-down creation of decision trees and a framework from which classification rules could be extracted from a decision tree as it was grown. In addition, facilities for user-selection of initial attribute selection were provided.

BLUE is not application specific, however, it was noted in Chapter 4 that discretizing numerically valued attributes into categorical ranges resulted in models that were not any more optimized or comprehensible with the utilization of BLUE. This indicates that BLUE is most useful for building models that are based primarily on categorical attributes.

Many studies that utilize machine learning techniques to induce decision trees and rules utilize a measure of accuracy of the created model to rate how good or bad a given algorithm performs. Ankerst (2000) points out that the motivating factor for the selection

of a decision tree algorithm to represent knowledge is based upon one of two reasons: one is for prediction purposes, the other is for description purposes. The motivating factors leading to this dissertation's outcome was to provide a framework from which domain and non-domain experts could build comprehensible models to be used for prediction and knowledge discovery purposes. Consequently, this study addressed both of Ankerst's (2000) motivating factors.

Implications

The work presented in this paper can be thought of as a first step toward the development of a visual data mining approach that uses one visual method for displaying local detail between relational database table attribute pairs and another visual method to provide a global context for the derivation of the model. The work demonstrates that BLUE can be utilized to interactively create decision trees and classification rule-sets, models that are comprehensible and can be easily implemented with computers.

The knowledge discovery field could benefit greatly from data visualization techniques such as BLUE. The volume of data that is being created in numerous application domains is exceeding the resources many companies possess in understanding the underlying relationships that exist in their data. Consequently, new visualization methods promise new tools from which non-domain expert users could help cull data and construct representative models that could be used for future classification or prediction purposes. Perhaps more importantly, researchers could possess a tool from which they could perform specific explorations. The limitation to the number of ways a data set could be viewed would only be limited by the imagination of the researcher. The primary

implication is that BLUE bridges the gap between the two types of users. Nevertheless, it would be more likely that a domain expert would uncover unknown information in the data than the non-expert user. This suggests that there are primarily two markets for BLUE.

One market consists of the expert users who wish to further explore a given data set or application domain. The other market could be thought of as the set of application domains that does not have the resources and/or time necessary to build models based upon the consultation of a domain expert. This group's primary concern might be with the construction of models by non-domain experts to help set pricing and inventory levels, for instance. The overriding implication is that BLUE is a useful technique for organizations that need to address the issue of mining relational databases with primarily categorically valued attributes. Consequently, BLUE could be applied to research-specific and practical problems alike.

Recommendations

There are a number of areas where BLUE could be utilized for additional study. Firstly, there is the possibility of modifying BLUE so that multivariate data could be accommodated. This could result in full association rule support. Next, the area of incremental learning could be addressed. Many data sets represent valid data but only for a moment in time. It would be useful to see how BLUE could be modified to accommodate incremental changes in its derived models. Another research area could involve BLUE's accommodation for attributes with continuous values. This would require consideration to the discretization of attribute ranges – Berchtold et al. (1998)

provide one approach for selecting such attribute ranges. Finally, BLUE could be modified in a manner so as to facilitate the inclusion/determination of missing values.

BLUE could also be used as a template for other visual data mining techniques. These techniques might include a visualization technique to view locally correlated data and a visualization method to represent the overall global model. Consideration might be given for the use of color for both visualization techniques. The first BIDS prototype was attempted with use of grayscale correlation rectangles but it was found that most images were too dark to precipitate a good analysis, hence, green and red colors were added.

Summary

Many application domains are growing exponentially in terms of database size. The ability to analyze these databases in an accurate but timely manner is becoming difficult. Consequently, these issues have led researchers in an effort to build models that represent meaningful relationships in the data sets. The purpose of building these models is two-fold. One purpose is to help predict future events based upon past performance; another is to discover relationships in the data that were unknown before the model was constructed.

Several data mining techniques have been developed to equip researchers with the ability to analyze and interpret relationships in data sets. The type of data mining that was addressed in this dissertation concerned classification. Classification models are most often presented in the form of decision trees and classification rules. In general terms, classification algorithms can be split into two major areas; information-theoretic approaches and statistical approaches.

Information-theoretic approaches rely on entropy measures to maximize information-gain, minimize uncertainty, and thus end up with subsets of data that facilitate homogeneous splitting of nodes in a decision tree. Statistical techniques, such as discriminant analysis, K-nearest neighbor, and Bayes rules, focus on unequal probability distributions and unequal misclassification costs. A limitation associated with most of these techniques is their use of fully-automated algorithms. These algorithms are typically non-interactive, hide the model derivation process from the user, require the assistance of a domain expert, are frequently application-specific, and do not always clearly translate detected relationships.

Possessing the ability to interactively guide the development of a model can be beneficial as it may enhance the researcher's ability to explore and understand various complex relationships within the data. Presently, the full promise of data mining's mission has not been realized. There is a need for the development of interactive visual data mining methods.

This paper introduced a data visualization algorithm, BLUE, as an alternative to present decision tree and classification rule construction methods. BLUE visually supports the process of classification. Based upon the visualization technique of BLUE's Independence Diagrams (BIDS), BLUE facilitates the interactive extraction of decision trees and classification rules from data sets. BIDS are based upon Berchtold et al. (1998) independence diagrams. BIDS differ from independence diagrams in that they represent certain data values with color, utilize a different tuple density measure, are focused on relational databases with primarily categorical attributes, and provide guidelines/algorithms to read the images.

BLUE utilizes the abilities of the human visual system to detect patterns and edges in images to direct the model derivation process. The emphasis on the visualization of data facilitates BLUE's use by domain and non-domain users alike. In addition, the algorithm employs a mechanism allowing the user to backtrack to previously visited nodes. This facilitates exploration of the data set as well as the capability to extract models in an intuitive manner.

A model derived with BLUE is represented with a decision tree. This model provides a global context upon which local detail can be utilized to direct the induction process. BIDS are utilized to view local detail. BIDS facilitate exploration of the data set and direct the induction of the decision tree model. As a decision tree is induced individual classification rules representing the different paths, from the root node to the leaf nodes, are extracted.

The process begins with the selection of an initial splitting node, or root node, of a decision tree. BLUE allows for any type of splitting strategy to be used for the induction of the initial node or subsequent nodes. This flexibility allows users who prefer to utilize an alternative splitting strategy, such as information theory or statistics, to induce portions of the decision tree or the entire model. This facility allows domain and non-domain experts the ability to use BLUE for explorative or predictive model building.

BLUE was empirically tested with five data sets representing a variety of application domains. Each data set was tested with three different methods of selecting the initial splitting attribute; these included gainratio, cardinality, and personal preference. The results of these experiments were compared in terms of number of leaf nodes/classification rules produced and model comprehensibility.

It was found that BLUE induced decision trees that were comparable in number of leaf nodes/classification rules with a personal preference splitting strategy when compared with those created with C4.5's gainratio approach or cardinality. In terms of comprehensibility, decision trees induced with personal preference for the initial splitting attribute induced models that were more expressive, comprehensible, and explorative when compared with those created with information theoretic and cardinality measures for the initial splitting attribute.

A second experiment was performed to determine if there was a correlation between initial splitting attribute and resultant support levels in classification rules induced with BLUE. Results of this experiment indicated that slightly higher support levels were required with a personal preference splitting technique when compared with gainratio and cardinality approaches. However, when standard deviation was taken into account, statistical significance differences were not exhibited.

The results of these experiments confirmed BLUE to be a viable approach for the induction of decision trees and their corresponding classification rules. The benefits of utilizing this technique are many-fold: models can be induced interactively, induced nodes can be backtracked upon facilitating explorative use by expert users, and multiple splitting strategies may be utilized facilitating any combination of selection of decision tree nodes. Consequently, there are many application domains where domain experts and non-domain experts could utilize BLUE to construct comprehensible models.

Appendix A

Zoo Data Set

This appendix presents the results of applying BLUE to the Zoo data set. This data set consists of 18 attributes (animal name, 15 boolean, and 2 numeric). There are 101 instances and no missing values. One of the numeric attributes is the number of legs an animal has. Discretization of this attribute followed the following mapping:

Number of animal legs	Categorical assignment in data set
0	1
2	2
4	3
5	4
6	5
8	6

The other numeric attribute is the classification attribute. It classifiers animals as follows:

Class	Set of Animals
1	aardvark, antelope, bear, boar, buffalo, calf, cavy, cheetah, deer, dolphin, elephant, fruitbat, giraffe, girl, goat, gorilla, hamster, hare, leopard, lion, lynx, mink, mole, mongoose, oppossum, oryx, platypus, polecat, pohy, porpoise, puma, pussycat, raccoon, reindeer
2	chicken, crow, dove, duck, flamingo, gull, hawk, kiwi, lark, ostrich, parakeet, penguin, pheasant, rhea skimmer, skua, sparrow, swan, vulture, wren

3	pitviper, seasnake, slowworm, tortoise, tuatara
4	bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna
5	frog, newt, toad
6	flea, gnat, honeybee, housefly, ladybird, moth, termite, wasp
7	clam, crab, crayfish, lobster, octopus, scorpion, seawasp, slug, starfish, worm

The first attribute in the data set was removed – it was an attribute that has unique values for each instance (i.e. name of each animal).

Zoo was analyzed with BLUE three different times, each with a different initial splitting attribute. First, the information-gain approach that C4.5 uses was selected. The results are presented in Table 5. This data set resulted in three attributes having the same GainRatio – the first attribute of the three, feathers, was chosen randomly from the three as the first splitting attribute.

The second splitting strategy is based upon a cardinality measure. The results are shown in Table 6.

The final splitting strategy was based upon user preference. In this case, Airborne was selected as a primary splitting attribute. The results of all three splitting strategies can be seen in Table 7. The three fully-grown decision trees can be seen in Figures 19, 20, and 21, respectfully.

Attribute	Entropy	InfoGain	SplitInfo	GainRatio
Hair	1.6	.7906	.984	.8035
*Feathers	1.6726	.718	.7179	1.0001
Eggs	1.5604	.8302	.9795	.8476
Milk	1.4162	.9744	.9743	1.0001
Airborne	1.921	.4696	.7911	.5936
Aquatic	2.001	.3896	.9397	.4146
Predator	2.2971	.0935	.9914	.0943
Toothed	1.5249	.8657	.9686	.8938
Backbone	1.7143	.6763	.6762	1.0001
Breathes	1.776	.6146	.7375	.8334
Venemous	2.1302	.2604	.3994	.6520
Fins	1.9239	.4667	.6538	.7138
Legs	1.0276	1.363	2.0337	.6702
Tail	1.8902	.5004	.8228	.6082
Domestic	2.3398	.0508	.5539	.0917
Catsize	2.0822	.3084	.988	.3121

Total Information in (T) = 2.3906
* Initial Splitting Attribute

Table 5: Summarized Gain Ratio Calculations (Zoo Data Set)

Attribute	Cardinality
*Hair	2
Feathers	2
Eggs	2
Milk	2
Airborne	2
Aquatic	2
Predator	2
Toothed	2
Backbone	2
Breathes	2
Venemous	2
Fins	2
Legs	6
Tail	2
Domestic	2
Catsize	2

* Initial Splitting Attribute

Table 6: Summarized Cardinality Calculations (Zoo Data Set)

Criteria	Number of Leaves/Rules
GainRatio	17
Cardinality	18
Personal Preference	14

Table 7: Summarized Results (Zoo Data Set)

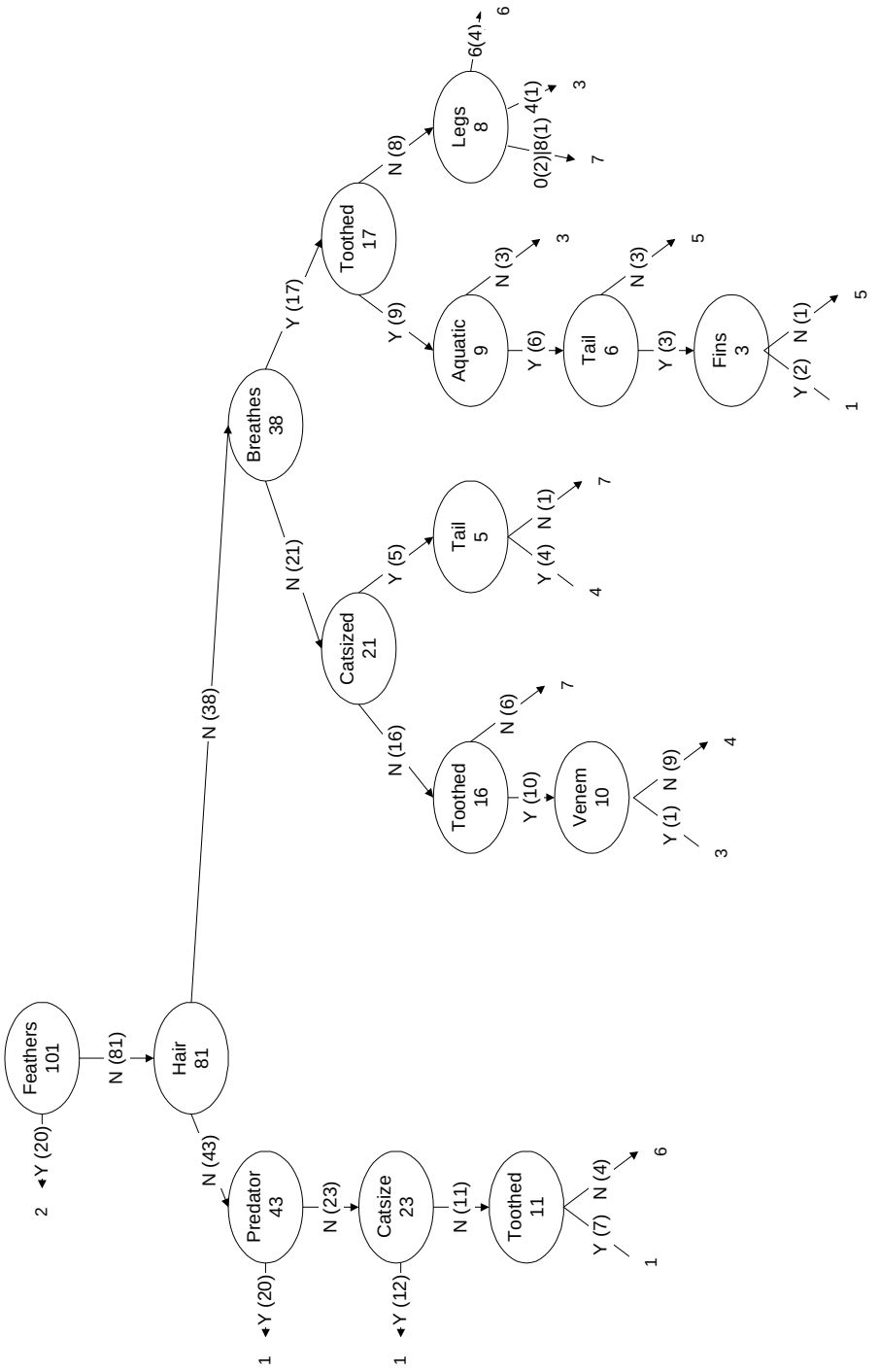


Figure 19: Fully Induced Decision Tree with Gain Ratio Criteria (Zoo Data Set)

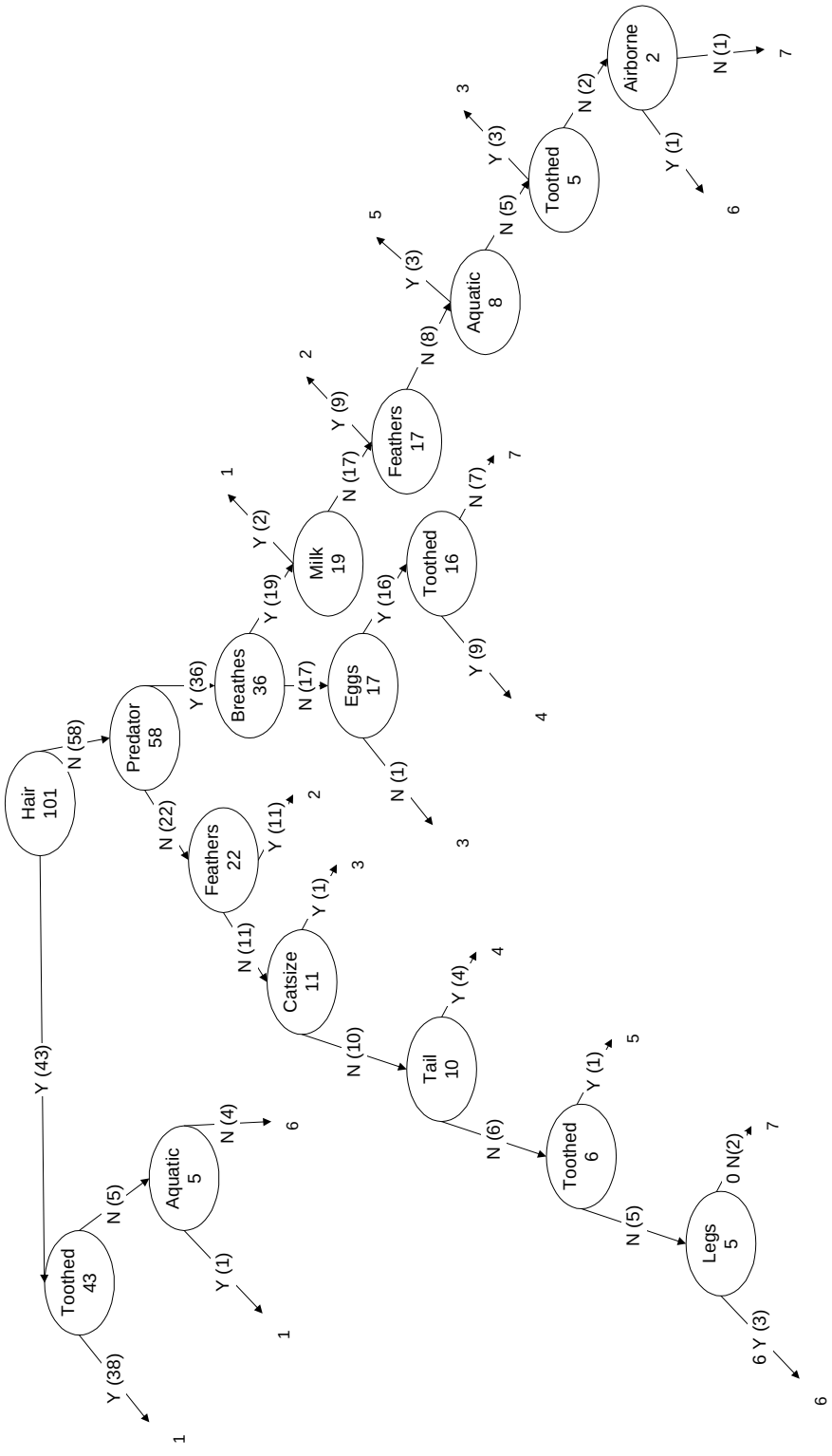


Figure 20: Fully Induced Decision Tree with Cardinality Criteria (Zoo Data Set)

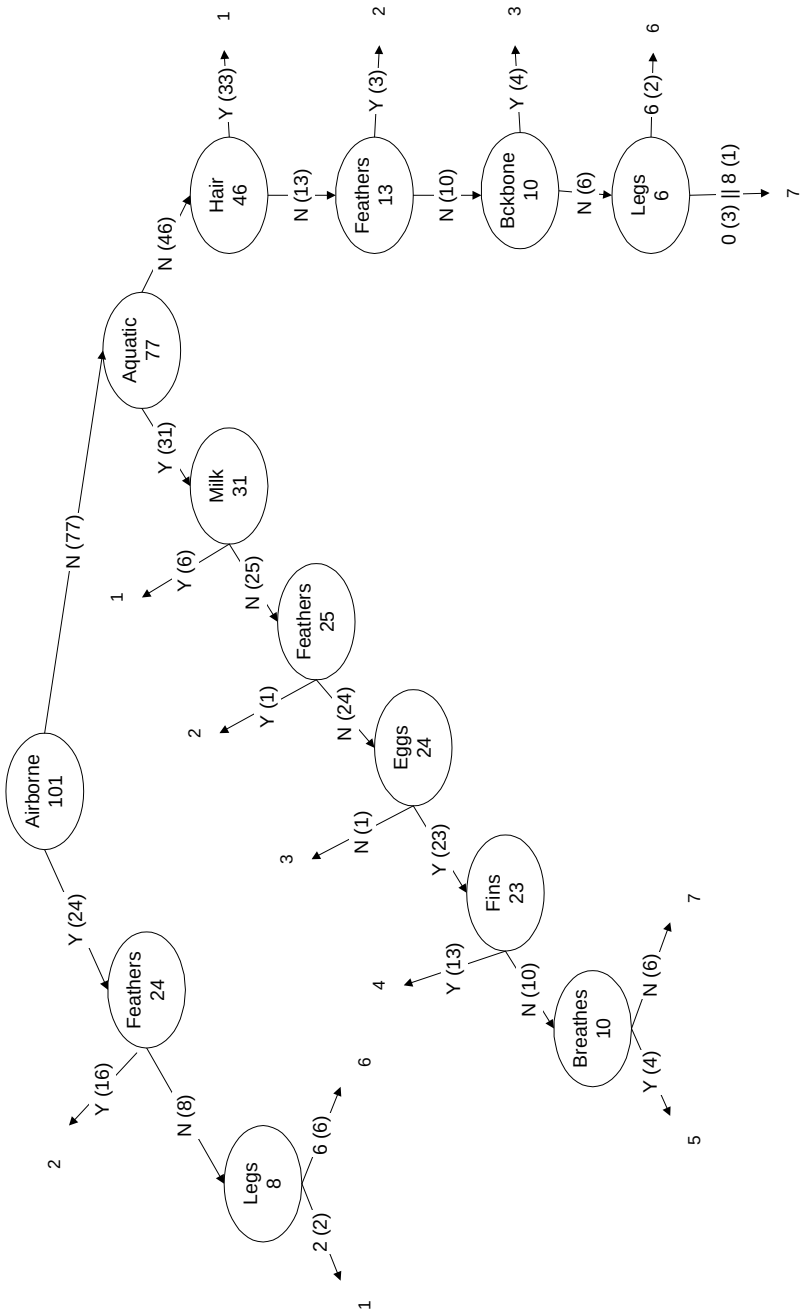


Figure 21: Fully Induced Decision Tree with Preference Criteria (Zoo Data Set)

Appendix B

Iris Data Set

This appendix presents the results of applying BLUE to the Iris data set. This data set consists of 5 attributes (4 numeric and one categorical). There are 150 instances and no missing values. The four numeric valued attributes are sepal length, sepal width, petal length, and petal width. Discretization of these attributes was necessary. The following mapping was used:

Attribute	Categorical assignment in data set		
	Setosa(0)	Versicolor(1)	Virginica(2)
Sepal length	4.3-5.4	5.5-6.1	6.2-7.9
Sepal width	2.0-2.7	2.8-3.2	3.3-4.4
Petal length	1.0-2.0	2.1-4.9	5.0-6.9
Petal width	0.0-.6	0.7-1.7	1.8-2.5

As indicated above, the three classification possibilities are Iris-Setosa, Iris-Versicolor, and Iris-Virginica.

Iris was analyzed with BLUE three different times, each with a different initial splitting attribute. First, the information-gain approach that C4.5 uses was selected. Petal Length possessed the highest GainRatio so it was selected as the initial splitting attribute. The results are shown in Table 8.

The second splitting strategy is based upon a cardinality measure. The results are shown in Table 9. In this case, all four non-classifying attributes have the same cardinality. As result, sepal length was randomly selected as a primary splitting attribute.

The final splitting strategy was based upon user preference. In this case, Sepal width was selected as a primary splitting attribute. The results of all three splitting strategies can be seen in Table 10. The three fully-grown decision trees can be seen in Figures 22, 23, and 24.

Attribute	Entropy	InfoGain	SplitInfo	GainRatio
Sepal Length	.9035	.6815	1.5773	.4321
Sepal Width	1.2791	.3059	1.4867	.2058
*Petal Length	.2603	1.3247	1.5819	.8374
Petal Width	.4037	1.1813	1.578	.7486

Total Information in (T) = 1.585
* Initial Splitting Attribute

Table 8: Summarized Gain Ratio Calculations (Iris Data Set)

Attribute	Cardinality
*Sepal Length	3
Sepal Width	3
Petal Length	3
Petal Width	3

* Initial Splitting Attribute

Table 9: Summarized Cardinality Calculations (Iris Data Set)

Criteria	Number of Leaves/Rules
GainRatio	11
Cardinality	11
Personal Preference	17

Table 10: Summarized Results (Iris Data Set)

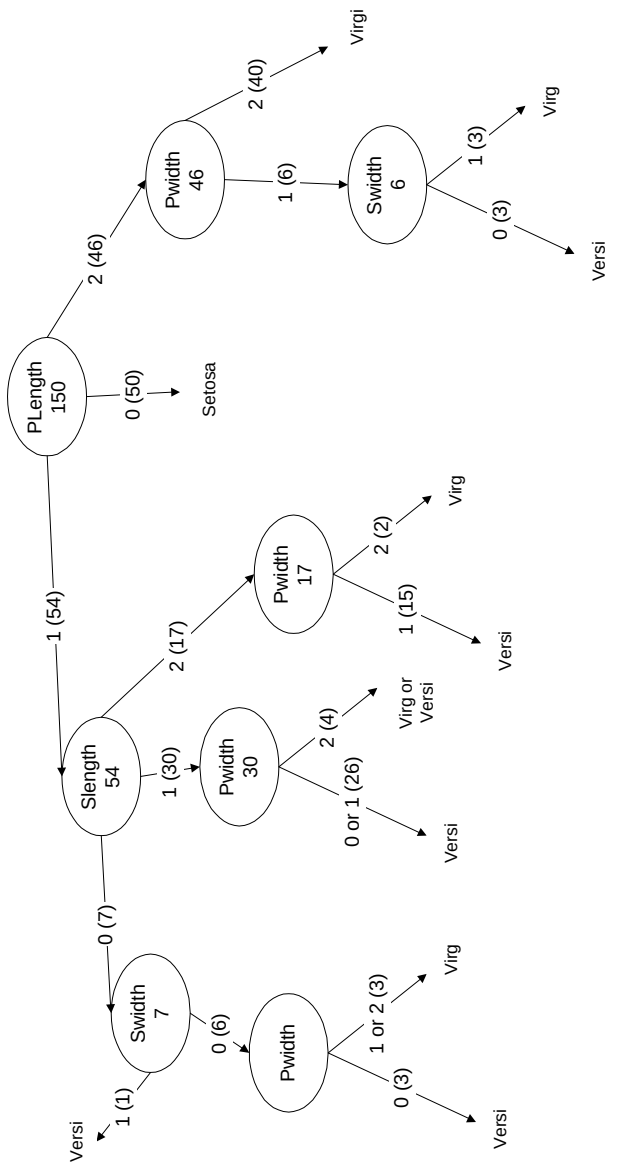


Figure 22: Fully Induced Decision Tree with Gain Ratio Criteria (Iris Data Set)

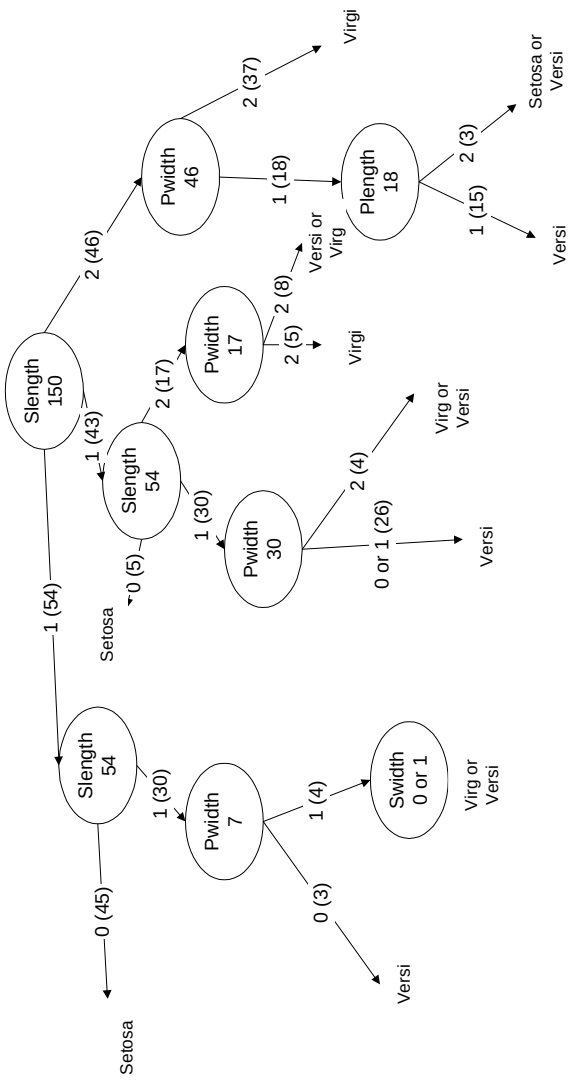


Figure 23: Fully Induced Decision Tree with Cardinality Criteria (Iris Data Set)

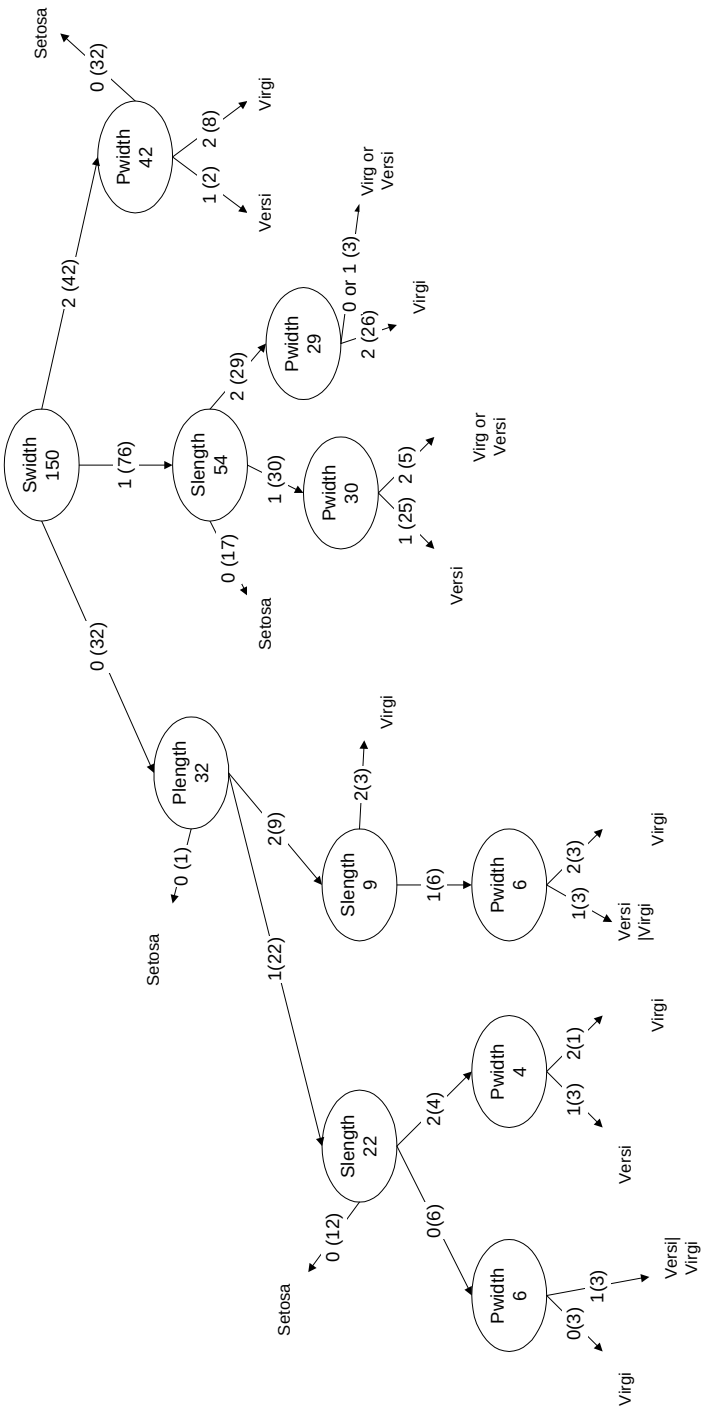


Figure 24: Fully Induced Decision Tree with Preference Criteria (Iris Data Set)

Appendix C

Glass Data Set

This Appendix presents the results of applying BLUE to the Glass data set. This data set consists of 12 attributes (10 numeric, and 1 categorical). There are 214 instances and no missing values. Nine of the numeric attributes needed to be split into categorical ranges. Discretization of these attributes followed the following mapping:

Attribute	Discretization ranges for numeric attributes		
	1	2	3
RI	1.51115-1.51687	1.51689-1.51841	1.51844-
Na	10.73-13.02	13.04-13.58	13.6-
Mg	0-.32	.33-3.59	3.6-
Al	.29-1	1.05-1.58	1.61-
Si	69.81-72.49	72.5-72.99	73-
K	0	.02-.76	.81-
Ca	5.43-6.96	7.08-11.64	12.24-
Ba	0	.06-3.15	
Fe	0	.01-.51	

The first attribute in the data set, Id number, was removed – it is an attribute that has unique values for each instance.

Glass was analyzed with BLUE three different times, each with a different initial splitting attribute. First, the information-gain approach that C4.5 uses was selected. The results are shown in Table 11. GainRatio is maximum for Barium (Ba), and so it was selected as the first splitting attribute.

The second splitting strategy is based upon a cardinality measure. The results are shown in Table 12. As can be seen from Table 12, two of the nine non-classifying attributes, Barium and Iron, have the same cardinality. As result, Iron (Fe) was randomly selected between the two as the primary splitting attribute.

The final splitting strategy was based upon user preference. In this case, Iron (Fe) was selected as a primary splitting attribute. The results of all three splitting strategies can be seen in Table 13. The three fully-grown decision trees can be seen in Figures 25, 26, 27, and 28.

Attribute	Entropy	InfoGain	SplitInfo	GainRatio
Refractive Index (RI)	1.972	.2046	1.5849	.1291
Sodium (Na)	1.8683	.3083	1.5836	.1947
Magnesium (Mg)	1.7933	.3833	1.4461	.2651
Aluminum (Al)	1.8405	.3361	1.3629	.2466
Silicon (Si)	2.0938	.0828	1.5847	.0522
Potassium (K)	1.7805	.3961	.8466	.4679
Calcium (Ca)	2.0439	.1327	.4574	.2901
*Barium (Ba)	1.8551	.3212	.6747	.4761
Iron (Fe)	2.1277	.0489	.9119	.0536

Total Information in (T) = 2.1766
* Initial Splitting Attribute

Table 11: Summarized Gain Ratio Calculations (Glass Data Set)

Attribute	Cardinality
Refractive Index (RI)	3
Sodium (Na)	3
Magnesium (Mg)	3
Aluminum (Al)	3
Silicon (Si)	3
Potassium (K)	3
Calcium (Ca)	3
Barium (Ba)	2
*Iron (Fe)	2

* Initial Splitting Attribute

Table 12: Summarized Cardinality Calculations (Glass Data Set)

Criteria	Number of Leaves/Rules
GainRatio	72
Cardinality	55
Personal Preference	55

Table 13: Summarized Results (Glass Data Set)

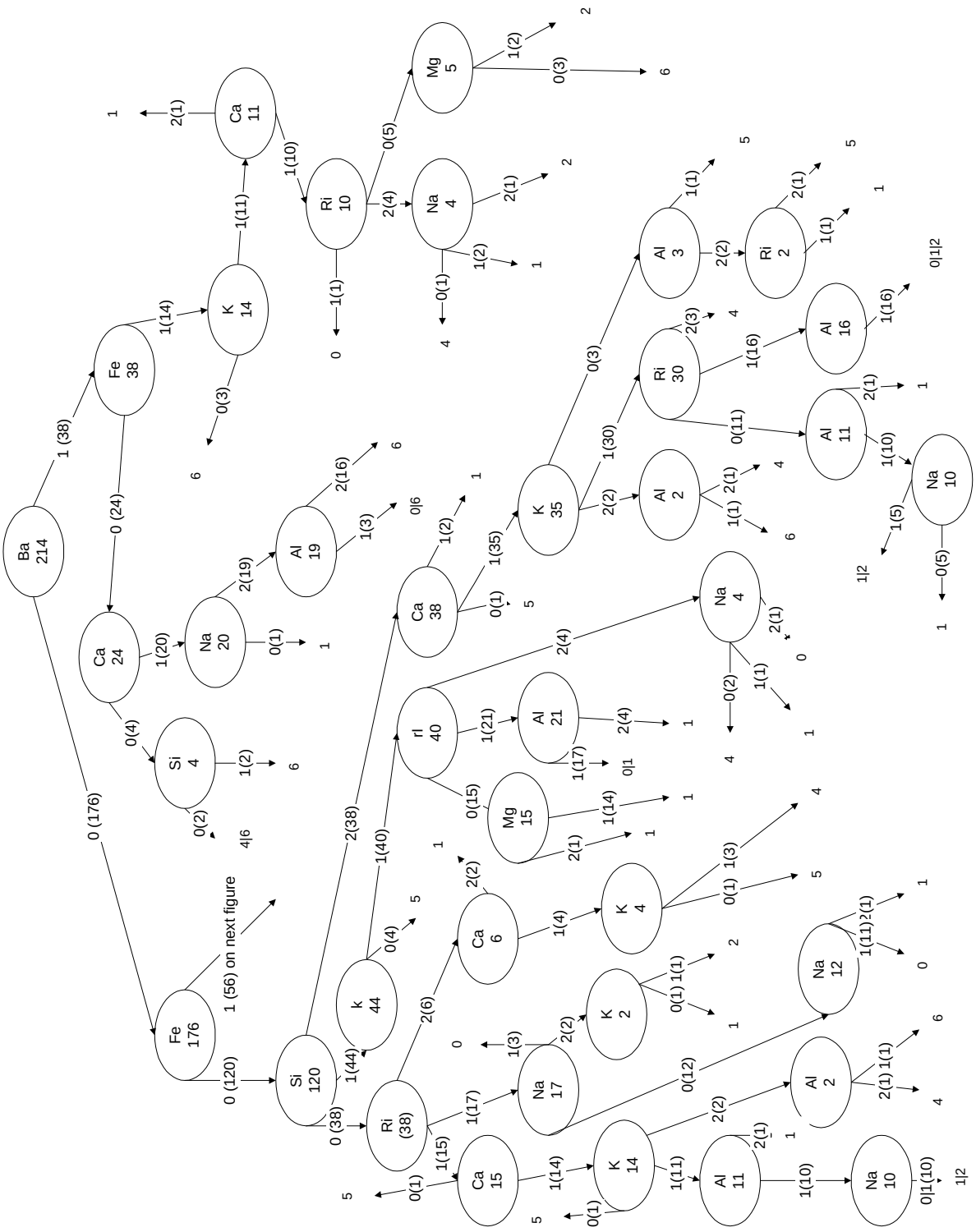


Figure 25: Fully Induced Decision Tree with Gain Ratio Criteria (Glass Data Set)

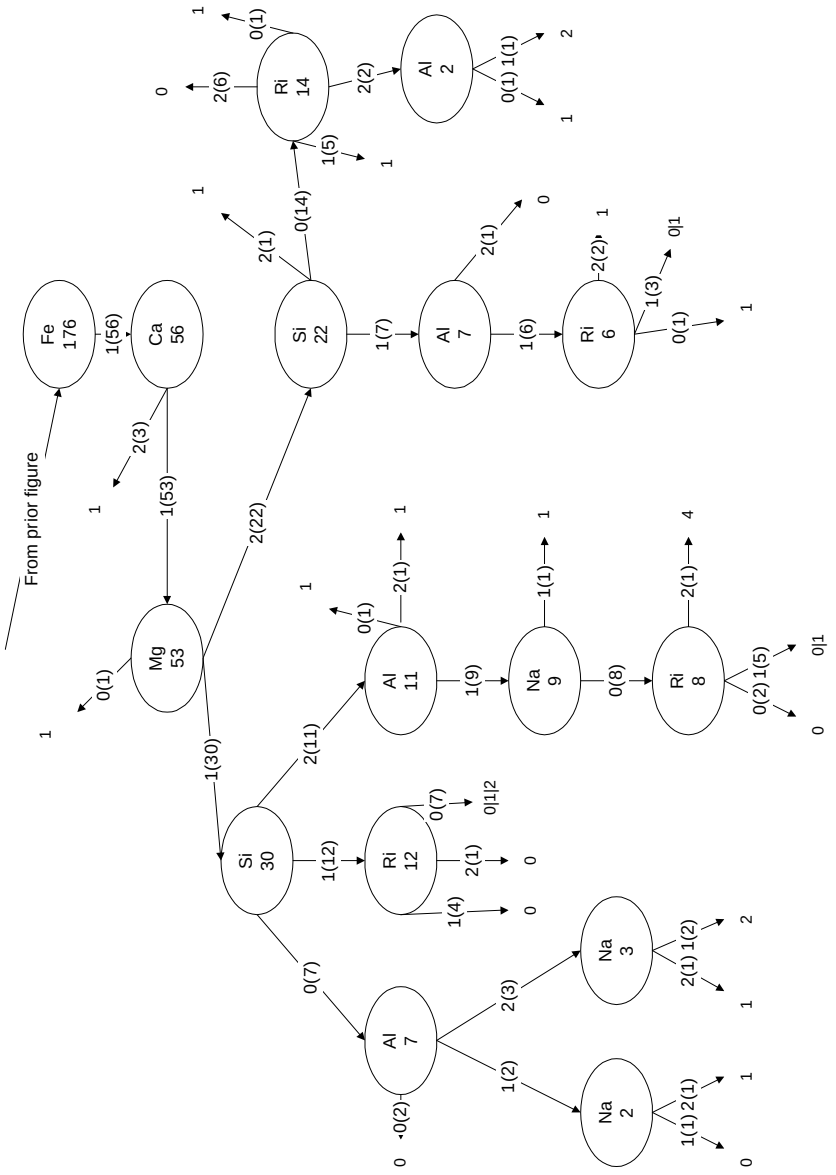


Figure 26: Fully Induced Decision Tree with Gain Ratio Criteria (Glass Data Set) (Continued)

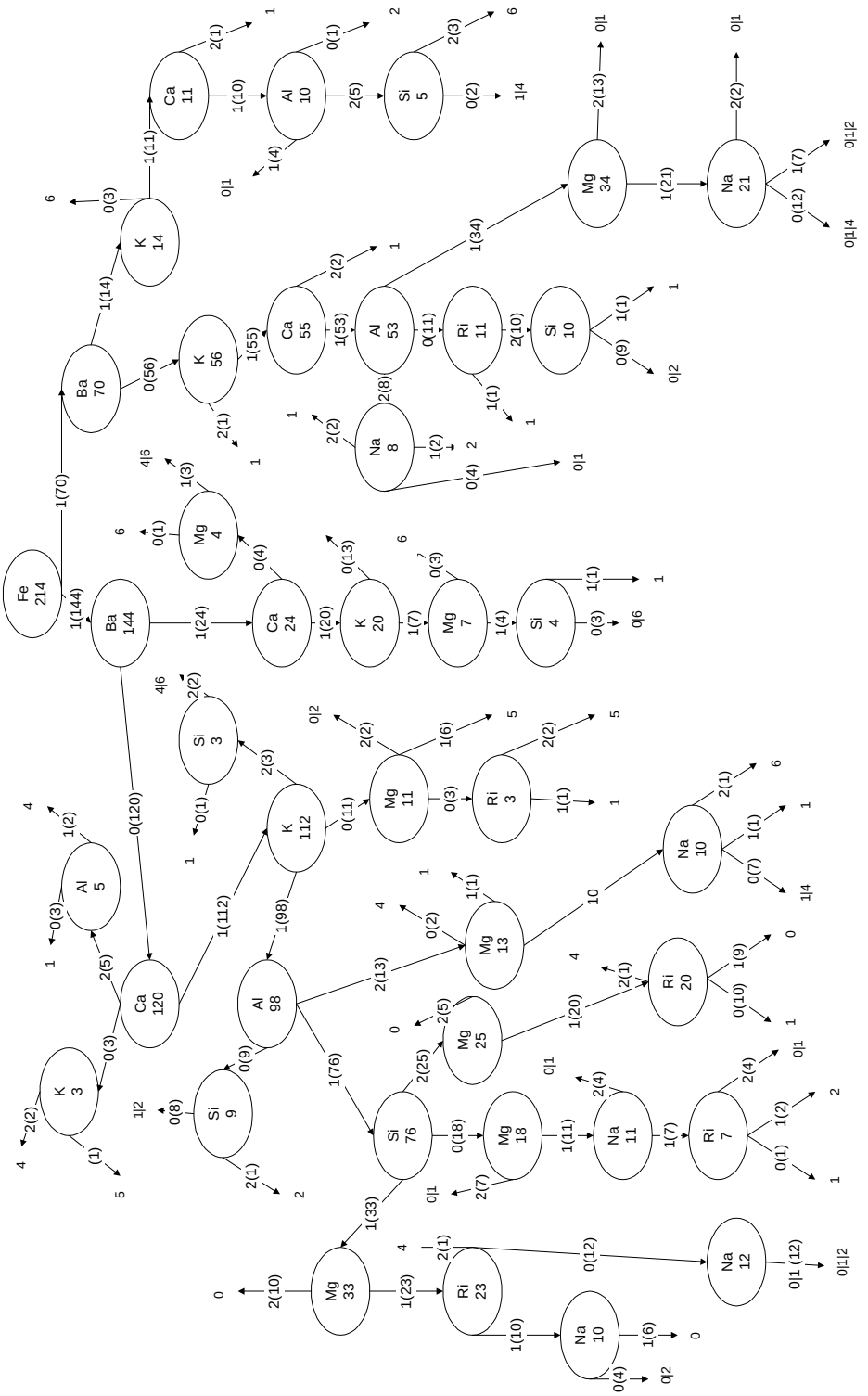


Figure 27: Fully Induced Decision Tree with Cardinality Criteria (Glass Data Set)

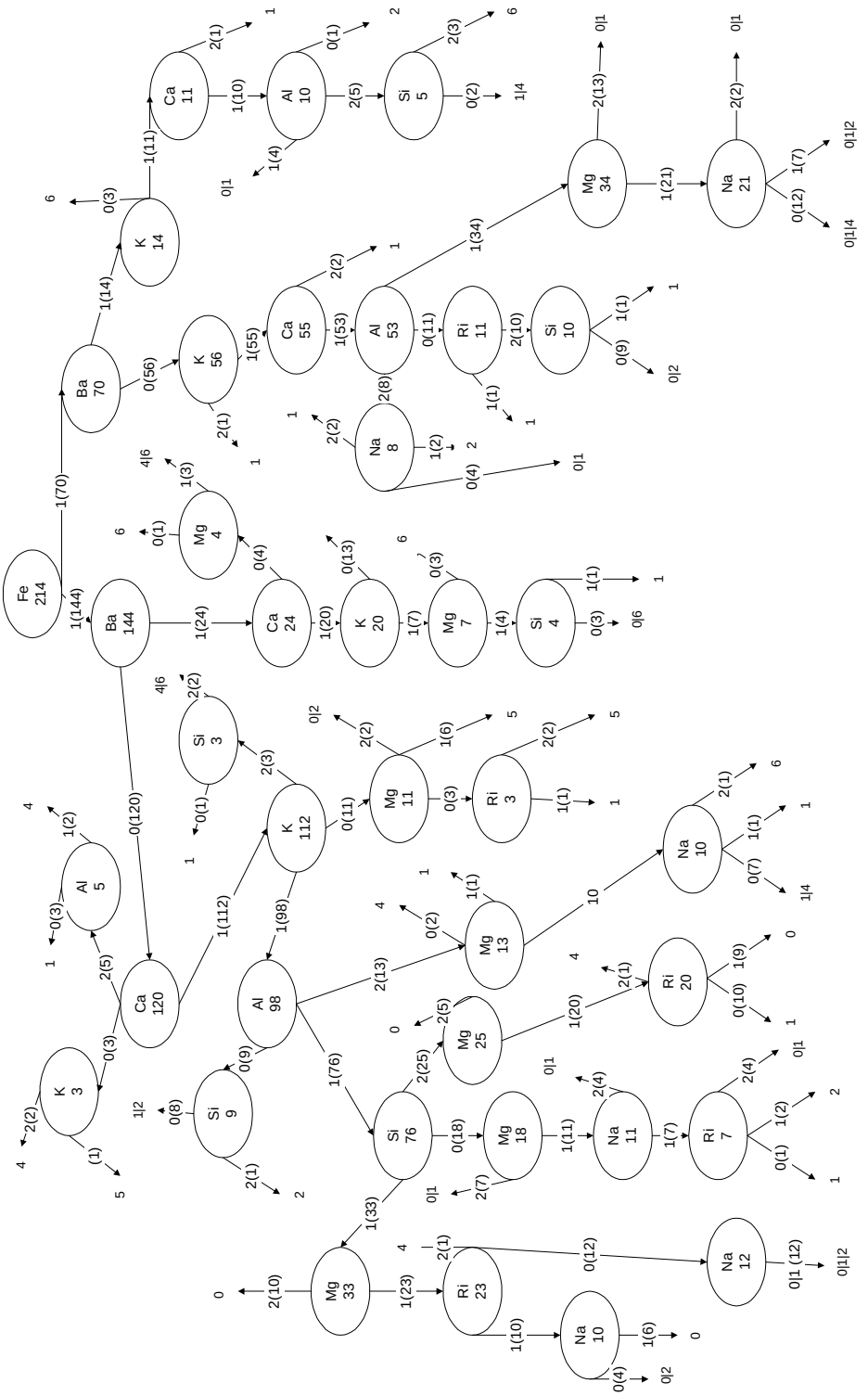


Figure 28: Fully Induced Decision Tree with Preference Criteria (Glass Data Set)

Appendix D

Voting Data Set

This appendix presents the results of applying BLUE to the voting data set. This data set consists of 17 attributes (all Boolean). There are 435 instances and no missing values.

Voting was analyzed with BLUE three different times, each with a different initial splitting attribute. First, the information-gain approach that C4.5 uses was selected. The results are shown in Table 14. Physician had the highest GainRatio so it was selected as the first splitting attribute.

The second splitting strategy is based upon a cardinality measure. The results are shown in Table 15. As can be seen from Table 15, all sixteen non-classifying attributes have the same cardinality. As a result, MX was randomly selected as a primary splitting attribute.

The final splitting strategy was based upon user preference. In this case, Education was selected as a primary splitting attribute. The results of all three splitting strategies can be seen in Table 16. The three fully-grown decision trees can be seen in Figures 29, 30, and 31.

Attribute	Entropy	InfoGain	SplitInfo	GainRatio
Handicapped Infants	.8362	.1261	1.1451	.1101
Water Project	.9619	.0004	1.3906	.0003
Adoption	.53	.4353	1.1184	.3892
*Physician	.4149	.5474	1.1256	.4863
El Salvador	.5399	.4224	1.1819	.3574
Religious	.8151	.1472	1.0878	.1353
Anti-Satellite	.7617	.2006	1.152	.1741
Aid to…	.6221	.3402	1.1657	.2918
MX	.6518	.3105	1.2383	.2507
Immigration	.9573	.005	1.1027	.0045
Synfuels	.8551	.1072	1.178	.0910
Education	.5881	.3742	1.2835	.2915
Superfund	.7345	.2278	1.259	.1809
Crime	.627	.3353	1.1747	.2854
Duty-Free	.7419	.2204	1.2659	.1741
Export Admin.	.8603	.102	1.323	.0771

Total Information in (T) = .9623
* Initial Splitting Attribute

Table 14: Summarized Gain Ratio Calculations (Voting Data Set)

Attribute	Cardinality
Handicapped Infants	2
Water Project	2
Adoption	2
Physician	2
El Salvador	2
Religious	2
Anti-Satellite	2
Aid to…	2
*MX	2
Immigration	2
Synfuels	2
Education	2
Superfund	2
Crime	2
Duty-Free	2
Export Admin.	2

* Initial Splitting Attribute

Table 15: Summarized Cardinality Calculations (Voting Data Set)

Criteria	Number of Leaves/Rules
GainRatio	35
Cardinality	53
Personal Preference	46

Table 16: Summarized Results (Voting Data Set)

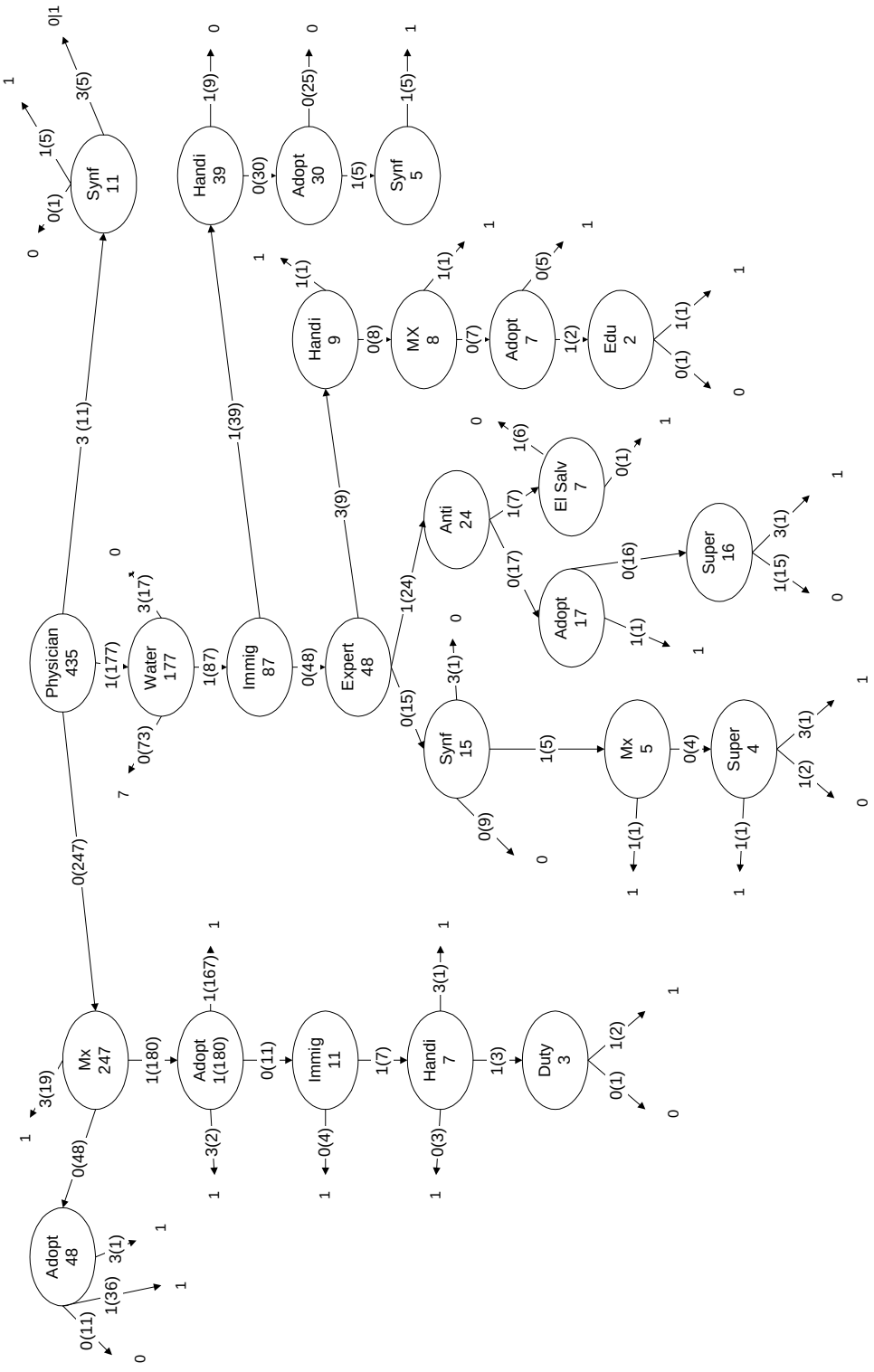


Figure 29: Fully Induced Decision Tree with Gain Ratio Criteria (Voting Data Set)

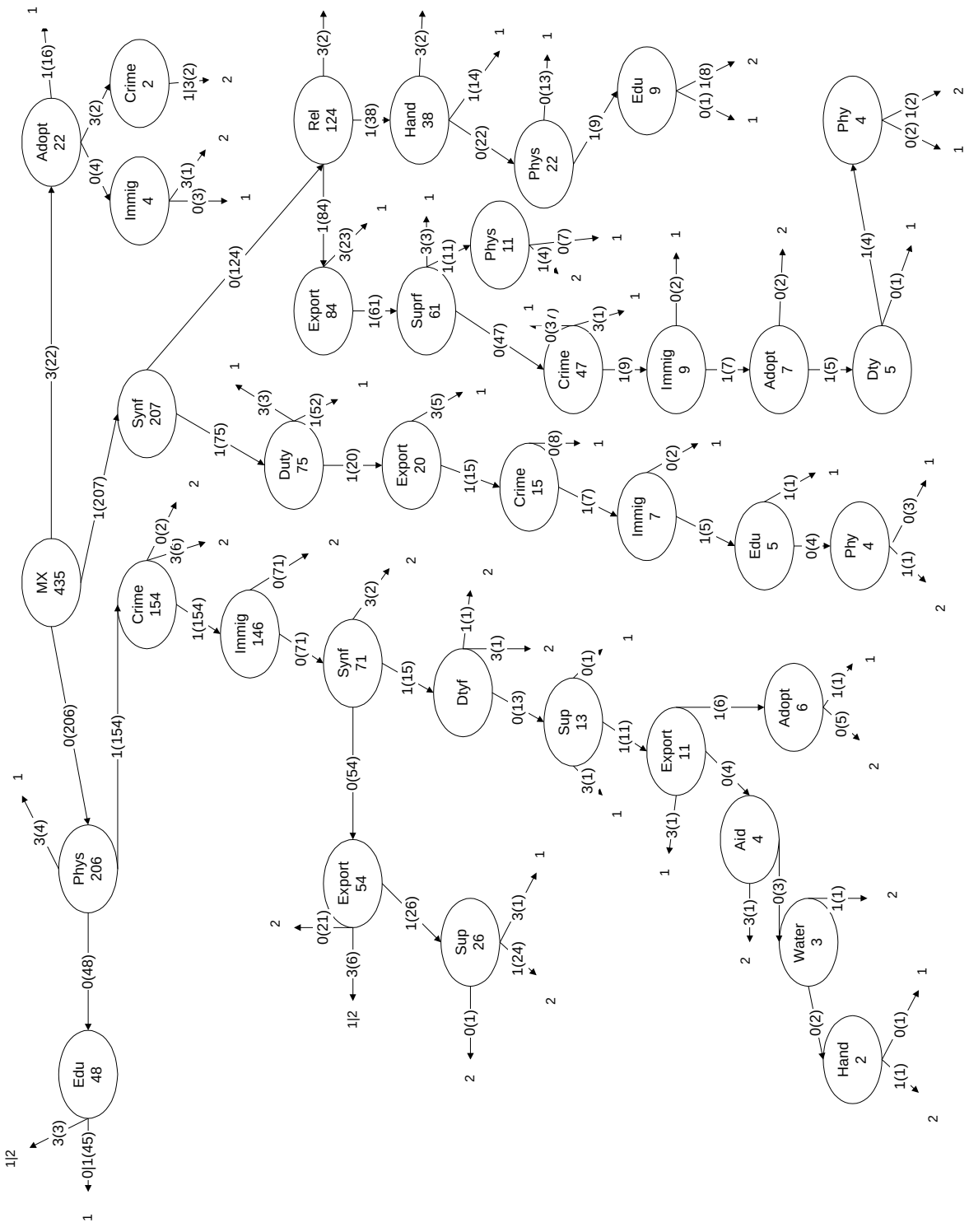


Figure 30: Fully Induced Decision Tree with Cardinality Criteria (Voting Data Set)

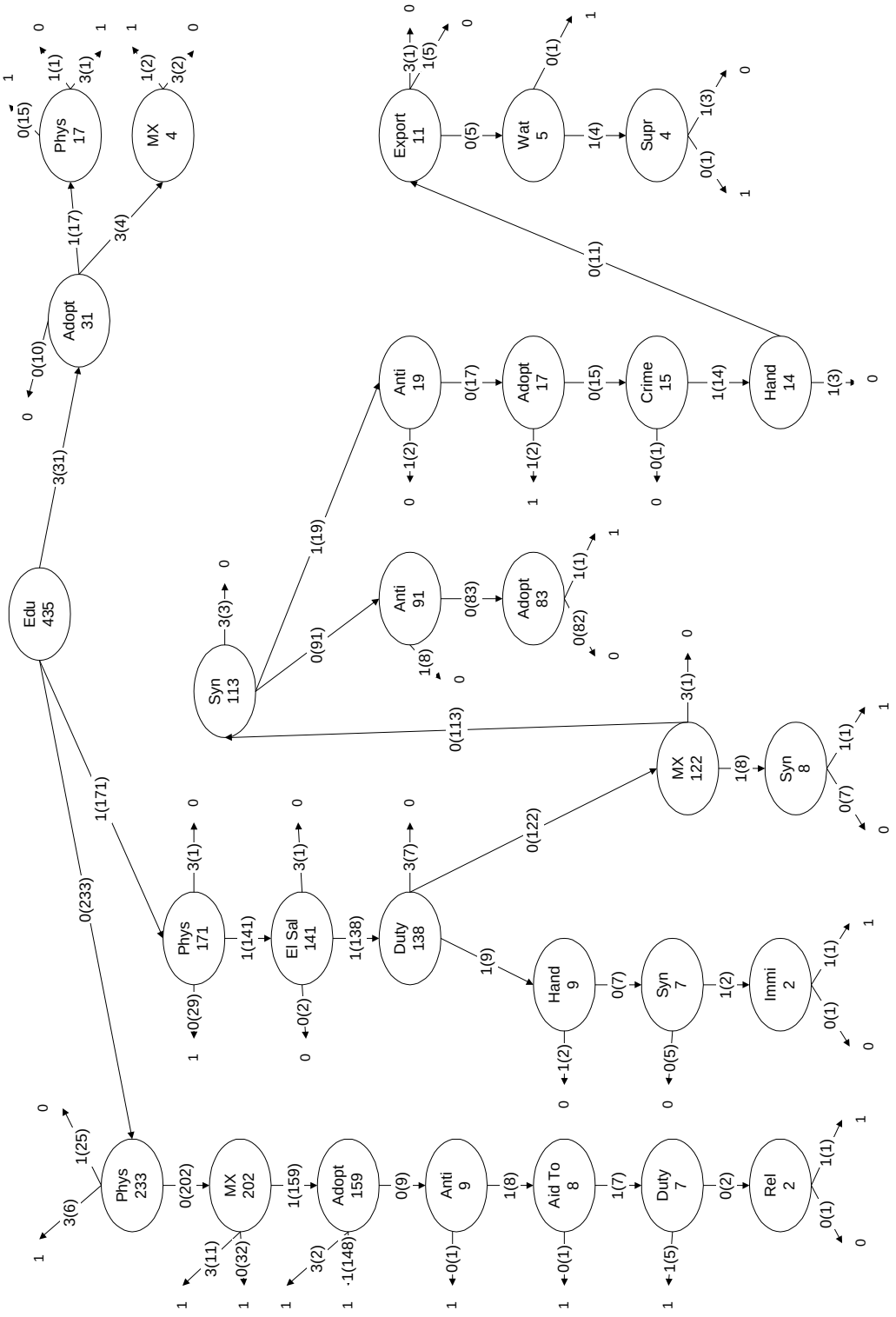


Figure 31: Fully Induced Decision Tree with Preference Criteria (Voting Data Set)

Appendix E

Titanic Data Set

This appendix presents the results of applying BLUE to the Titanic data set. This data set consists of 4 attributes (all categorical in nature). There are 2201 instances and no missing values.

Titanic was analyzed with BLUE three different times, each with a different initial splitting attribute. First, the information-gain approach that C4.5 uses was selected. The results are shown in Table 17. This data set resulted in Sex having the highest gain-ratio, hence, it was selected as the first splitting attribute.

The second splitting strategy is based upon a cardinality measure. The results are shown in Table 18. As can be seen from Table 18, two of the non-classifying attributes have the same cardinality. As result, Age was randomly selected as a primary splitting attribute.

The final splitting strategy was based upon user preference. In this case, Class was selected as the primary splitting attribute. The results of all three splitting strategies can be seen in Table 19. The three fully-grown decision trees can be seen in Figures 32, 33, and 34.

Attribute	Entropy	InfoGain	SplitInfo	GainRatio
Class	.8484	.0593	1.8441	.0322
Age	.0912	.0065	.2844	.0229
*Sex	.3715	.5362	.7482	.7167

Total Information in (T) = .907
* Initial Splitting Attribute

Table 17: Summarized Gain Ratio Calculations (Titanic Data Set)

Attribute	Cardinality
Class	4
*Age	2
Sex	2

* Initial Splitting Attribute

Table 18: Summarized Cardinality Calculations (Titanic Data Set)

Criteria	Number of Leaves/Rules
GainRatio	24
Cardinality	24
Personal Preference	22

Table 19: Summarized Results (Titanic Data Set)

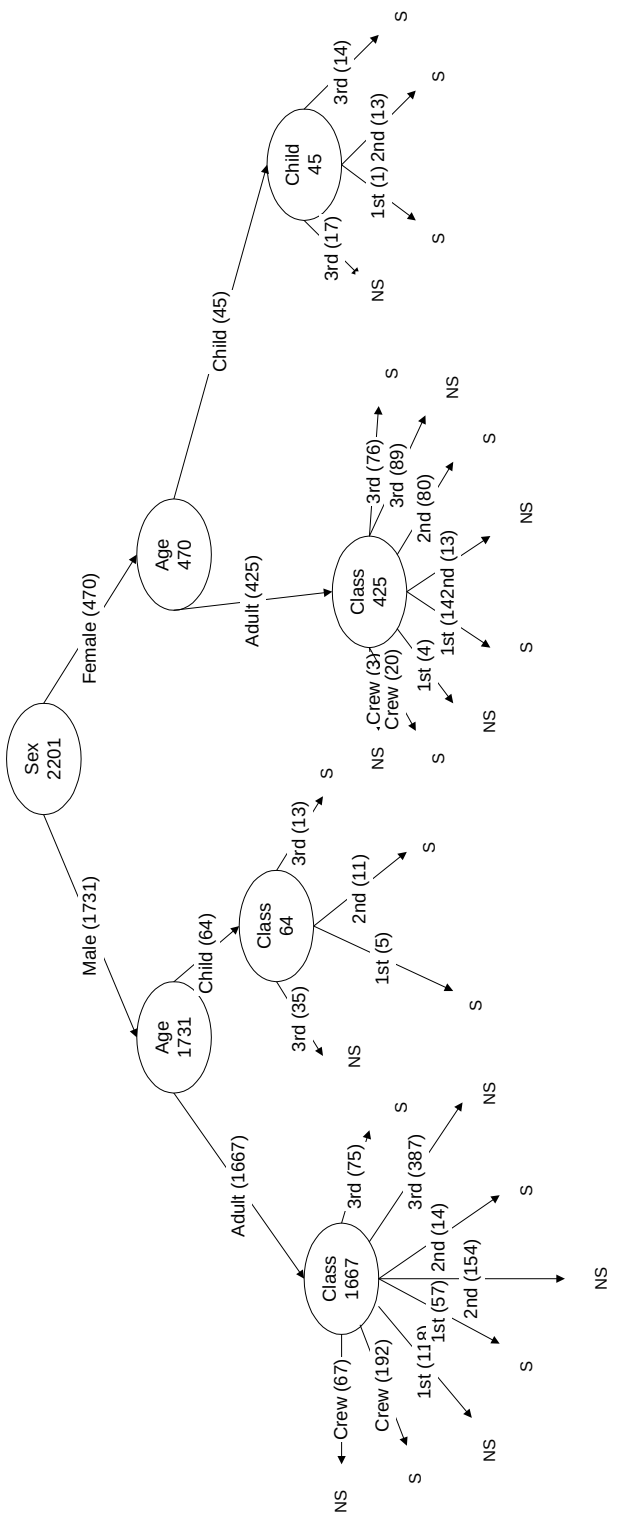


Figure 32: Fully Induced Decision Tree with Gain Ratio Criteria (Titanic Data Set)

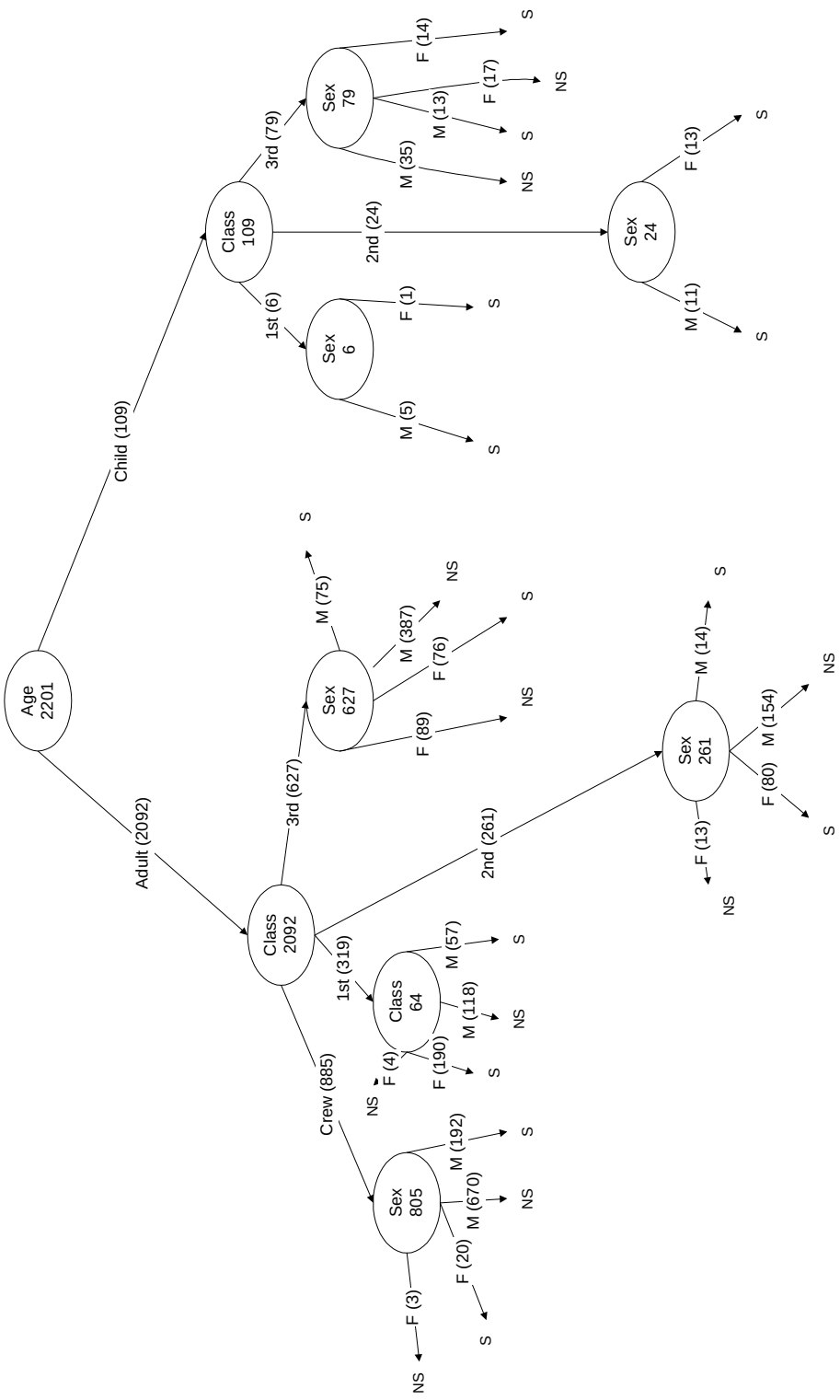


Figure 33: Fully Induced Decision Tree with Cardinality Criteria (Titanic Data Set)

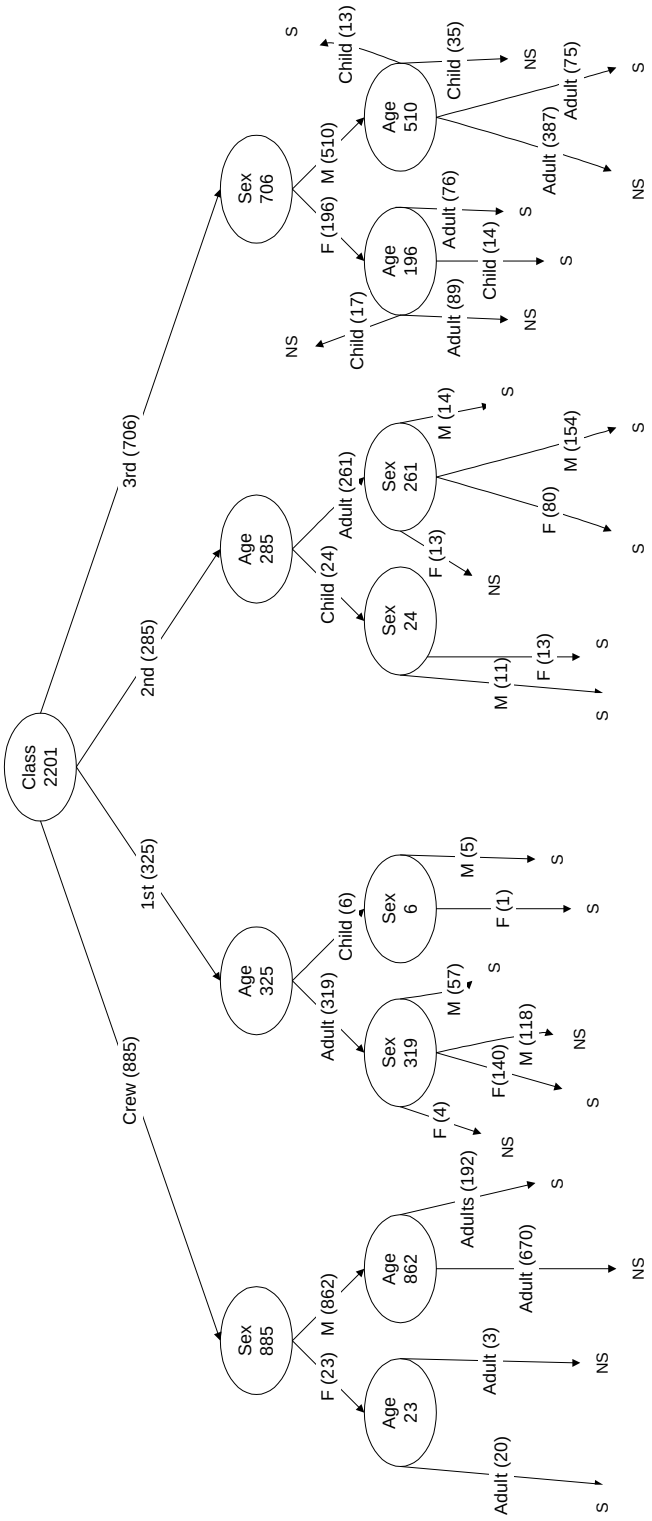


Figure 34: Fully Induced Decision Tree with Preference Criteria (Titanic Data Set)

Reference List

Agrawal, R., Ghosh, S., Imielinski, T., Iyer, B. & Swami, A. (1992). An interval classifier for database mining applications. *Proceedings of the 18th International Conference on Very Large Data Bases,* 560-573. Vancouver, Canada.

Agrawal, R., Imielinski, T., & Swami, A. (1993a). Database mining: a performance perspective. *IEEE Transactions on Knowledge and Data Engineering,* 5(6), 914-925.

Agrawal, R., Imielinski, T., & Swami, A. (1993b). Mining association rules between sets of items in large databases. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data,* 207-216. Washington, D.C.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceedings of the 20th International Conference on Very Large Data Bases*, 487-499. Santiago, Chile.

Agrawal, R., & Srikant, R. (1995). *Mining generalized association rules. Proceedings of the 21st International Conference on Very Large Data Bases,* 407-419. Zurich, Switzerland.

Ankerst, M., Keim, D. A., & Kriegel, H. P. (1996). Circle segments: a technique for visually exploring large multidimensional data sets. *Proceedings of the IEEE 1996 Conference on Visualization.* San Francisco, CA.

Ankerst, M., Elsen, C., Ester, M., & Kriegel, H. P. (1999). Visual classification: an interactive approach to decision tree construction. *Proceedings of the Fifth ACM SIGKKD International Conference on Knowledge Discovery and Data Mining,* 392-396. San Diego, CA.

Ankerst, M. (2000). *Visual Data Mining*, Ph.D. Thesis, University of Munich, Germany.

Bayardo Jr., R. J., & Agrawal, R. (1999). Mining the most interesting rules. *Proceedings of the Fifth ACM SIGKKD International Conference on Knowledge Discovery and Data Mining,* 145-154. San Diego, CA.

Berchtold, S., Jagadish, H. V., & Ross, K. A. (1998). Independence diagrams: a technique for visual data mining. *Proceedings of the Fourth ACM SIGKKD International Conference on Knowledge Discovery and Data Mining*, 139-143. New York, NY.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Pacific Grove, CA: Wadsworth and Brooks/Cole Advanced Books and Software.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123-140.

Chen, M. S., Han, J., & Yu, P. S. (1996). Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering, 8*(6), 866-883.

Chi, E., Riedl, J., Barry, P., & Konstan, J. (1998). Principles for information visualization spreadsheets. *IEEE Computer Graphics and Applications, 18*(4), 30-38.

Cohen, E., Datar, M., Fujiwara, S., Gionis, A., Indyk, P., Motwani, R., Ullman, J. D., & Yang, C. (2000). Finding interesting associations without support pruning. *Proceedings of the 16th Annual IEEE Conference on Data Engineering*, 489-499. San Diego, CA.

Cox, K. C., Eick, S. G., Wills, G. J., & Brachman, R. J. (1997). Visual data mining: recognizing telephone calling fraud. *Journal of Data Mining and Knowledge Discovery, 1*(2), 225-231.

Dietterich, T. G. (1990). *Readings in Machine Learning.* San Mateo, CA: Morgan Kaufmann Publishers.

Emmons, E. E., Jennings, M. J., & Edwards, C. (1999). An alternative classification method for northern Wisconsin lakes. *Canadian Journal of Fisheries and Aquatic Sciences, 56*(4), 661-669.

Fayyad, U. M., & Irani, K. B.(1992). The attribute selection problem in decision tree generation. *Proceedings of the Tenth National Conference on Artificial Intelligence*, 104-110. San Jose, CA.

Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM, 39*(11), 27-34.

Fortner, G. (1995). Column Data. In P. Green (Ed.), *The Data Handbook* (Vol. 2, pp. 100-108). Columbia, MD: Springer-Verlag.

Frank, E., & Witten, I. H. (1998). Using a permutation test for attribute selection in decision trees. *Proceedings of the 15th International Conference on Machine Learning*, 152-160. Madison, WI.

Freitas, A. A. (2000). Understanding the crucial differences between classification and discovery of association rules. *SIGKDD Explorations, 2*(1), 65-69.

Freund, Y., & Shapire. (1996). Experiments with a new boosting algorithm. *Proceedings of the 13th International Conference on Machine Learning*, 148-156. San Francisco, CA.

Friedman, J. (1977). A recursive partitioning decision rule for nonparametric classification. *IEEE Transactions on Computers, C-26*(4), 404-408.

Friendly, M. (1995). Conceptual and visual models for categorical data. *The American Statistician*, 49, 153-160.

Furnas, G. W. (1982). *The FISHEYE view: A new look at structured files*. Bell Laboratories Technical Memorandum, #82-11221-22.

Gehrke, J., Ramakrishnan, R., Ganti, V. (1998). RainForest – a framework for fast decision tree construction of large datasets. *Proceedings of the 24th International Conference on Very Large Data Bases*, 127-162. New York, NY.

Gershon, N., & Eick, S. G. (1998). Guest editors' introduction to information visualization - the next frontier. *Journal of Intelligent Information Systems, 11*, 199-204.

Goodman, R. M., & Smyth, P. (1988). Decison tree design from a communication theory standpoint. *IEEE Transactions on Information Theory, 34*(5), 979-994.

Han, J., Cai, Y., & Cercone, N. (1992). Knowledge discovery in databases: an attribute-oriented approach. *Proceedings of the 18th International Conference on Very Large Data Bases*, 547-559. Vancouver, BC.

Han, J., Fu, Y., Wang, W., Chiang, J., Gong, W., Koperski, K., Li, D., Lu, Y., Rajan, A., Stefanovic, N., Xia, B., & Zaiane, O. R. (1996a). DBMiner: a system for mining knowledge in large relational databases. *Proceedings of the Second ACM SIGKKD International Conference on Knowledge Discovery and Data Mining*, 250-255. Portland, OR.

Han, J., Huang, Y., Cercone, N., & Fu, Y. (1996b). Intelligent query answering by knowledge discovery techniques. *IEEE Transactions on Knowledge and Data Engineering, 8*(3), 373-390.

Han, J., Chee, S., & Chiang, J. Y. (1998). Issues for on-line analytical mining of data warehouses. *Proceedings of the 1998 ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, 2*, 1-5. Seattle, WA.

Hunt, E. B., Marin, J., & Stone, P. J. (1966). *Experiments in Induction*. New York: Academic Press.

Iizuka, Y., Shiohara, H., Iizuka, T., & Isobe, S. (1998). Automatic visualization method for visual data mining. *Proceedings of the Second Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 173-185. Melbourne, Australia.

Imielinksi, T., & Mannila, H. (1996). A database perspective on knowledge discovery.

Communications of the ACM, 39(11), 58-64.

Jerding, D., & Stasko, J. (1998). The information mural: a technique for displaying and navigating large information spaces. *IEEE Transactions on Visualization and Computer Graphics, 4*(3), 257-271.

Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., & Verkamo, A.I. (1994). Finding interesting rules from large sets of discovered association rules. *In Proceedings of the 3rd International Conference on Information and Knowledge Management*, 401-407. Gaithersburg, MD.

Lewis II, P. M. (1962). The characteristic selection problem in recognition systems. *IRE Transactions on Information Theory*, IT-8(2), 171-178.

Liu, B., Hsu, W., & Ma, Y. (1998). Integrating classification and association rule mining. *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, 80-86. New York, NY.

Marmelstein, R. E. (1999). *Evolving Compact Decision Rule Sets*, Ph.D. Thesis, Air Force Institute of Technology, Dayton, OH.

Mehta, M., Rissanen, J, & Agrawal, R. (1995). MDL-based decision tree pruning. *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*, 216-221. Menlo Park, CA.

Mehta, M., Agrawal, R., & Rissanen, J. (1996). SLIQ: a fast scalable classifier for data mining. *Proceedings of the 5th International Conference on Extending Database Technology*, 18-32. Avignon, France.

Mitchell, T. M. (1997). *Machine Learning.* New York: McGraw-Hill.

Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (Editors). (1994). *Machine Learning, Neural and Statistical Classification*. New York: Ellis Horwood.

Murthy, K. V. S. (1997). *On Growing Better Decision Trees from Data*, Ph.D. Thesis, The Johns Hopkins University, Baltimore, Maryland.

Park, J. S., Chen, M. S., & Yu, P. S. (1995). An effective hash based algorithm for mining association rules. *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, 175-186. San Jose, CA.

Piatetsky-Shapiro, G. (1991a). Discovery, analysis, and presentation of strong rules. In G. Piatetsky-Shapiro and W. J. Frawley (Eds.), *Knowledge Discovery in Databases* (pp. 229-248). Menlo Park, CA: AAAI Press / The MIT Press.

Piatetsky-Shapiro, G., & Frawley, W. J. (Editors). (1991b). *Knowledge Discovery in*

Databases: *An Overview*. Menlo Park: AAAI Press /MIT Press.

Quinlan, J. (1986). Induction of decision trees. *Machine Learning, 1*, 81-106.

Quinlan, J. (1987). Simplifying decision trees. *International Journal of Man-Machine Studies, 27*, 221-234.

Quinlan, J., & Rivest, R. L. (1989). Inferring decision trees using the minimum description length principle. *Information and Computation, 80*, 227-248.

Quinlan, J. R. (1993). *C4.5 - Programs for Machine Learning*. San Mateo, California: Morgan Kaufmann Publishers.

Rastogi, R., & Shim, K. (1998). PUBLIC: a decision tree classifier that integrates building and pruning. *Proceedings of 24th International Conference on Very Large Data Bases*, 404-415. New York, NY.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica, 14*, 465-471.

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics, 21*(3), 660-674.

Sarkar, M., & Brown, M. H. (1992). Graphical fisheye views of graphs. *Proceedings of Human Factors in Computing Systems*, 83-91. Monterey, CA.

Schapire, R. (1990). The strength of weak learnability. *Machine Learning, 5*(2), 197-227.

Shafer, J., Agrawal, R., & Mehta, M. (1996). SPRINT: a scalable parallel classifier for data mining. *Proceedings of 22nd International Conference on Very Large Data Bases*, 544-555. Bombay, India.

Shannon, C. (1948). A mathematical theory of communication. *Bell Systems Technical Journal, 27*, 379-423 and 623-656.

Shneiderman, B. (1994). Dynamic queries for visual information-seeking. *IEEE Software, 11*(6), 70-77.

Silberschatz, A., & Tuzhilin, A. (1996). What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering. Special Issue on Data Mining, 8*(6), 970-974.

Sun, J. (2000). Cardinality-based quantitative rule derivation. *Proceedings of IFIP International Conference on Intelligent Information Processing (16th IFIP World Computer Congress)*, 23-30. Beijing, China.

Tufte, E. R. (1990). *Envisioning Information*. Cheshire, CT: Graphics Press.

Wang, K., Zhou, S., & He, Y. (2000). Growing decison trees on support-less association rules. In T. Terano, H. Liu, & A.L.P. Chen (Eds.), *Knowledge Discovery and Data Mining Current Issues and New Applications* (pp 265-269). Columbia, MD: Springer-Verlag.

White, A. P., & Liu, W. Z. (1994). Bias in information-based measures in decision tree induction. *Maching Learning, 15*(3), 321-329.

Wu, X. (1995). *Knowledge Acquisition from Databases*. Greenwich, CT: Ablex Publishing Corporation.

Wu, X., & Urpani, D. (1999). Induction by attribute elimination. *IEEE Transactions on Knowledge and Data Engineering, 11*(5), 805-812.

Yoda, K., Fukuda, T., Morimoto, Y., Morishita, S., & Tokuyama, T. (1997). Computing optimized rectilinear regions for association rules. *Proceedings of the Third ACM SIGKKD International Conference on Knowledge Discovery and Data Mining,* 96-103. Newport Beach, CA.