MODEL SELECTION FOR LOGISTIC REGRESSION VIA
ASSOCIATION RULES ANALYSIS

A Dissertation in
Statistics
by
Pannapa Changpetch

© 2012 Pannapa Changpetch

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2012
The dissertation of Pannapa Changpetch was reviewed and approved* by the following:

Dennis Lin  
Distinguished Professor of Statistics and Supply Chain Management  
Dissertation Advisor  
Chair of Committee

Steven Arnold  
Professor of Statistics

Yu Zhang  
Assistant Professor of Statistics

Harriet Black Nembhard  
Professor of Industrial Engineering

Bruce Lindsay  
Willaman Professor of Statistics  
Head of the Department of Statistics

*Signatures are on file in the Graduate School
ABSTRACT

In this research, we develop a model selection procedure for logistic regression by implementing association rules analysis. This is particularly important for applications in which interactions are present. Interaction effects are very common in reality, but they have received little attention in the logistic regression literature. This is especially true for higher-order interactions. Here, we develop a model selection framework to address this problem. We do this by (1) exploring the combinations of input variables which have significant impacts to response (via association rules analysis); (2) selecting the potential (low- and high-order) interactions; (3) converting these potential interactions into new dummy variables; and (4) performing variable selections among all the input variables and the newly created dummy variables (interactions) to build up the optimal logistic regression model. The key advantages of the proposed framework include the ability to deal with a large number of interactions, the capacity to select potential interactions, and the provision of alternate setups for interactions. In the proposed method, interactions are incorporated into the logistic regression model. Higher-order interactions may also be included in the model, a feature not available in other approaches to logistic regression modeling.

Since we apply association rules analysis as a major technique for our model selection procedure, we study it thoroughly and discover major issue for this technique. Association rules analysis is a powerful methodology for exploring relationships among items in the form of rules. It has been widely used in a number of research studies and practices. In many applications, association rules analysis has been used to capture the sets of rules forming the model that represents a given dataset. Little is known, however, that the identical dataset can be represented via different sets of rules. The idea of equivalent models has not been explored. Accordingly, many studies investigating the goodness of model can be misleading because they focus exclusively on the particular underlined model of rules. Any comparison of techniques based on the extent to which each captures the true model will be invalid if such equivalence is not taken into account. With the overall objective of fostering accurate results in the field, we explore this major issue in a thorough manner and demonstrate that equivalent models can exist for any
dataset. In addition, we propose the benchmark model as a representative in explaining the relationships among variables for all equivalent models.

Next, we present a comprehensive analysis of and comparison between rules selection criteria. When association rules analysis was introduced in the early 1990s, the two basic criteria were used which we also apply with the proposed method. However, there are a lot more rules selection criteria available today. The challenge is to determine the criteria most appropriate to the proposed method in practice. The goal is to establish the character and properties of rules selection criteria that are robust for a wide variety of models for the scenarios of interest, i.e., those that involve (higher-order) interactions. Our methodological framework makes rules selection comparisons for the purpose of providing a fair comparison and thorough analysis. The advantages of our comparison framework are as follows. First, our model can compare rules selection criteria in order to search for all the possible true models not only for particular models or for specific rules. Second, the performance comparison is based on the simulation results from general datasets generated from the scenarios of interest rather than on specific datasets. Third, we do not depend on the judgments of experts for this study. Fourth, as it plays an important part in the comparison and analysis, the issue of equivalent models is taken seriously in this study. We propose this comparison framework as a benchmark for comparison procedures in future studies. Throughout the methodical simulation processes, the performance comparisons and recommendations for rules selection criteria are provided.

Further, we demonstrate that proposed method is compatible not only with the logistic regression model but with categorical models in general. The ability to search for optimal models for categorical models with more levels of response is also investigated. We demonstrate our model’s compatibility with the more complicated models using applications for multinomial logistic regression. Thus, the effectiveness and efficiency of our proposed method is confirmed.

This research is motivated by data mining technique, specifically market basket analysis. We present an approach that has never been tried before: specifically, we apply association rules analysis in order to incorporate interactions into logistic regression modeling. The issue of equivalent models which we discover in our study is shown to have a major impact on association rules analysis such that comparisons and analyses of any techniques or rules selection criteria that do not account for this issue are likely to be misleading and invalid. We present a comprehensive study of the rules selection criteria that are most practical and robust for our
proposed method. We extend the applications to multinomial logistic regression and confirm that the proposed method is not only practical for use with the logistic regression model but with categorical models in general.
# Table of Contents

List of Figures ................................................................................................................................ ix

List of Tables .................................................................................................................................. xi

Acknowledgments ........................................................................................................................ xiii

Chapter 1: Introduction ................................................................................................................... 1

Chapter 2: Logistic Regression via Association Rules Analysis .................................................... 6

2.1 Logistic Regression and Association Rules Analysis ............................................................... 6

2.2 The Proposed Method .............................................................................................................. 10

2.3 Illustrated Example Using the Proposed Method ...................................................................... 14

2.3.1 Comparisons Based on Simulated Datasets .................................................................. 18

2.4 Application: Lymphography Dataset ...................................................................................... 20

2.5 Discussion and Conclusion ..................................................................................................... 23

Chapter 3: Identifiability for Association Rules Analysis ............................................................... 25

3.1 An illustrative example .............................................................................................................. 26

3.2 Definitions and Generation of Equivalent Models ................................................................. 29

3.2.1 Definitions ......................................................................................................................... 29

3.2.2 Generating Equivalent Models ........................................................................................ 30

3.3 Theoretical Properties ............................................................................................................. 35

3.4 Representative for the Equivalent Models ............................................................................. 37

3.5 Revisit MONK’s dataset ......................................................................................................... 39

3.5.1 Equivalent Models for MONK’s dataset ........................................................................ 40

3.5.2 Dominant Model for MONK’s Dataset ........................................................................... 41

3.6 Conclusions ............................................................................................................................... 43

Chapter 4: Analysis and Comparison for Rules selection Criteria ................................................... 44
4.1 Rules Selection Criteria .................................................................................................. 45
4.2 Simulation Process and Comparison Procedure ............................................................. 51
  4.2.1 Details of the Design Matrix ................................................................................... 51
  4.2.2 Simulation Process .................................................................................................. 51
  4.2.3 Comparison Procedure ............................................................................................ 59
  4.2.4 Illustrated Example for Simulation Process ............................................................ 59
4.3 Performance Comparison and Analysis ......................................................................... 63
  4.3.1 The Comparison for Dependency on Support and Rule Size in Breaking Ties ...... 64
  4.3.2 Comparison of the Effects of Increases in the Variables ......................................... 66
  4.3.3 Results Summary .................................................................................................... 72
  4.3.4 Results for Simulations with Noise ......................................................................... 73
4.4 Conclusions .................................................................................................................... 74

Chapter 5: Multinomial Logit Models Selection via Association Rules Analysis ............... 75
  5.1 Multinomial Logistic Regression Model and Association Rules Analysis ................ 76
  5.2 Proposed Method with the Multinomial Logit Model .................................................. 77
  5.3 Application: Alligator Food Choice Dataset ............................................................... 80
  5.4 Conclusions .................................................................................................................... 82

Chapter 6: Conclusion and Future Work ...................................................................................... 83

Appendices .................................................................................................................................... 86
  Appendix A: Bar Charts for Comparing the Number of Rules Needed With and Without the
              Involvement of Support and Rule Size to Break Ties ....................................................... 87
  Appendix B: Bar charts for comparing the number of rules needed for
              input matrix with 6 binary variables and input matrix with 10 binary variables .......... 92
  Appendix C: Simulation Results Summary .............................................................................. 97
List of Figures

Figure 2-1: Framework for the proposed model building.................................................. 10
Figure 2-2: Workflow of the proposed method ................................................................. 13
Figure 3-1: Dominant Model vs. Disjoint Model ............................................................. 30
Figure 3-2: Model A vs. Model F .................................................................................... 382
Figure 3-3: Generation of disjoint model........................................................................... 42
Figure 3-4: Decision chart for the search procedure for the dominant model ............... 38
Figure 3-5: Procedure for obtaining the dominant model for Model R ....................... 42
Figure 4-1: Data generation procedure for two-rule model ........................................... 52
Figure 4-2: The performance testing procedure for each criterion ............................... 56
Figure 4-3: The simulation process ................................................................................ 58
Figure A1: Bar chart representing the number of rules needed for Model 1 ................. 88
Figure A2: Bar chart representing the number of rules needed for Model 2 ................. 88
Figure A3: Bar chart representing the number of rules needed for Model 3 ............... 89
Figure A4: Bar chart representing the number of rules needed for Model 4 ............... 89
Figure A5: Bar chart representing the number of rules needed for Model 5 ............... 90
Figure A6: Bar chart representing the number of rules needed for Model 6 ............... 90
Figure A7: Bar chart representing the number of rules needed for Model 7 ............... 91
Figure A8: Bar chart representing the number of rules needed for Model 8 ............... 91
Figure B1: Bar chart representing the number of rules needed for Model 1 ............... 93
Figure B2: Bar chart representing the number of rules needed for Model 2 ............... 93
Figure B3: Bar chart representing the number of rules needed for Model 3 ............... 94
Figure B4: Bar chart representing the number of rules needed for Model 4 ............... 94
Figure B5: Bar chart representing the number of rules needed for Model 5 ............... 95
Figure B6: Bar chart representing the number of rules needed for Model 6 ............... 95
Figure B7: Bar chart representing the number of rules needed for Model 7 ..................... 96
Figure B8: Bar chart representing the number of rules needed for Model 8 ..................... 96
Figure D1: Number of rules needed for Model 1 when noise is 10%
   for cosine, implication index, least contradiction, Kulczynski index, and PS............... 102
Figure D2: Number of rules needed for Model 2 when noise is 10%
   for cosine, implication index, least contradiction, Kulczynski index, and PS............... 103
Figure D3: Number of rules needed for Model 3 when noise is 10%
   for cosine, implication index, least contradiction, Kulczynski index, and PS............... 104
Figure D4: Number of rules needed for Model 4 when noise is 10%
   for cosine, implication index, least contradiction, Kulczynski index, and PS............... 105
Figure D5: Number of rules needed for Model 5 when noise is 10%
   for cosine, implication index, least contradiction, Kulczynski index, and PS............... 105
Figure D6: Number of rules needed for Model 6 when noise is 10%
   for cosine, implication index, least contradiction, Kulczynski index, and PS............... 107
Figure D7: Number of rules needed for Model 7 when noise is 10%
   for cosine, implication index, least contradiction, Kulczynski index, and PS............... 107
Figure D8: Number of rules needed for Model 8 when noise is 10%
   for cosine, implication index, least contradiction, Kulczynski index, and PS............... 109
List of Tables

Table 2-1: Attributes for MONK’s dataset ......................................................................................... 14
Table 2-2: Variable class results for MONK’s dataset ........................................................................ 15
Table 2-3: Candidate models for MONK’s dataset ............................................................................ 16
Table 2-4: Classical model for MONK’s dataset .................................................................................. 17
Table 2-5: Comparison between the proposed model and the classical model for MONK’s dataset .................................................................................................................................. 18
Table 2-6: Comparison between the proposed model and the classical models ................................. 19
Table 2-7: Responses and attributes for the lymphography dataset .................................................. 20
Table 2-8: Candidate models for the lymphography dataset ............................................................. 22
Table 2-9: Comparison between the proposed model and the classical models (lymphography) .................................................................................................................................. 23
Table 3-1: Dataset for model A ........................................................................................................... 26
Table 3-2: Illustrated equivalent models ............................................................................................ 28
Table 3-3: Dominant model of class 1 for MONK’s dataset (Model M) ............................................. 40
Table 3-4: Equivalent model for MONK’s dataset (Model R) ............................................................. 40
Table 4-1: Rules Selection Criteria summarized from Tan et. al 2004, Geng et al. 2006, and Blanchard et. al 2009. (order alphabetically) .................................................................................. 47
Table 4-2: The 14 initial models for the two-rule model ................................................................. 53
Table 4-3: The final models of our interest ....................................................................................... 55
Table 4-4: Data generation for design matrix 2(6) using Model 1 .................................................... 59
Table 4-5: Generated rules and their rankings for the confidence criterion ..................................... 61
Table 4-6: Generated rules and their rankings for the J-measure criterion ....................................... 62
Table 4-7: Are there any changes in the number of rules needed with and without support and rule size? ........................................................................................................... 64
Table 4-8: Three groups and subgroups based on their performances in the first comparison ........................................................................................................................................ 65

Table 4-9: Is there any change in the number needed between the first design matrix (2(6)) and second design matrix (2(10))? ........................................................................................................................................ 66

Table 4-10: Criteria based on three conditions ............................................................................................................................................................ 68

Table 5-1 Variable class results for the dataset ............................................................................................................................................................ 78

Table 5-2: Responses and attributes for the alligator food choice dataset ........................................................................................................................................ 80

Table C1: Number of rules needed for the design matrix 2(6) using scheme 1 .......... 98

Table C2: Number of rules needed for the design matrix 2(6) using scheme 2 ............ 99

Table C3: Number of rules needed for the design matrix 2(10) using scheme 2 ........ 100
Acknowledgments

I owe more than I can say to my dad, my mom, and my sister. I could not have reached this point in my academic journey without their unconditional love and support. Thank you so much dad, mom, and Mod for always being there for me.

I gratefully thank my advisor, Dr. Dennis Lin, for his great advice and wonderful support at every step of the way. His guidance has helped me tremendously not only in terms of my research but also in regard to my teaching and my professional life as a whole.

I would also like to thank my committee members, Dr. Steven Arnold, Dr Yu Zhang and Dr. Harriet Nembhard, for their consistent support and constructive criticism over the years. And, thanks to Thap Panitanarak for helping me with the simulations in this research.

Finally, I would like to thank the faculties in the Department of Statistics for the part they have played in the uniquely wonderful experience I have had at Penn State.
Chapter 1

Introduction

Logistic regression models are among the most important models for dichotomous responses. It has been used to explore relationships among variables, utilizing explanatory variables to predict response classes. Logistic regression models are used to make predictions and explain relationships among variables in a wide variety of areas, including, engineering, business, agriculture, and biomedical studies. However, selecting a logistic regression model when many main effects and interactions are involved is problematic. Incorporating interactions into models becomes challenging when the number of main effects increases, because the number of interactions between them grows at an accelerated rate. The classical method practically neglects interaction effects and selects models that include only main effects. In this research, we propose a systematic procedure that considers the interactions among variables, thereby producing models that are superior to those developed conventionally.

In logistic model building, different types of interactions have been treated in different ways. In general, interactions both between quantitative factors and interactions and between quantitative and categorical factors are considered during the process. However, the interactions among the categorical factors have been treated in a range of dissimilar ways. Some assume that there are no interactions between categorical variables, and therefore they are omitted from logistic regression models (Agresti, 2002). Other studies include these interactions in the model, but only after their main effects have been included (Hosmer and Lemeshow, 1989). In doing so, they build a model that includes only significant main effects in the first stage and consider only the interactions among them in the second stage (the hierarchy structure). In this research, we study the possible benefits of including these interactions in the logistic regression model by considering them concurrently with the main effects. The difficulty stems not only from determining which variables interact with each other, but also in deciding how to set up the interactions, given the wide range of options available when different levels of categorical variables are present.
To help us deal with the problem of interactions among variables, we turn to association rules analysis—a methodology for exploring relationships among items (or variables) in the form of rules. The basic idea driving association rules analysis is straightforward: it is to determine the rules important to a given dataset. These rules need to meet some given constraints, for example, a minimum support and a minimum confidence. One particularly popular version is market basket analysis, which is used to understand relationships between products and consumer purchasing behavior. A large number of studies make use of association rules analysis in a wide variety of areas: biology (Becquet et al. 2002; Calders et al. 2006; Kuo et al. 2007) business and marketing (Changchien et al. 2001; Wang et al. 2005; Shih et al. 2010), geography (Zhong et al. 1999; Appice et al. 2003; Lee et al. 2007), agriculture (Matsumoto 1998; Mclver et al. 2002), education (Garcia et al. 2007; Garcia et al. 2007; Merceron and Yacef 2008), photography (Ding et al. 2002; Tesic et al. 2003; Liu et al. 2008), economics (Dopfer and Potts 2004; Mossong et al. 2008), and so on. We apply association rules analysis to aid in the selection of potential interactions among categorical variables from a large pool of possibilities. Using this method, we are able to narrow the field of possible combinations by eliminating interactions with a low probability of contributing to the fit of the logistic model.

The first part of this research introduces the proposed method for the model selection procedure for logistic regression. Our model selection framework improves on the ability of classical model building to consider potential interactions. Association rules analysis streamlines the selection of important rules, converts those selected rules into interaction variables, and determines the optimal model for logistic regression by implementing a subset selection method that considers all the main effects and potential interactions. The key advantages of the proposed framework include the ability to deal with a large number of interactions, the capacity to select potential interactions, and the provision of alternate setups for interactions. In the proposed method, interactions are incorporated into the logistic regression model. Higher-order interactions can also be included in the model—a feature not available in other approaches to logistic regression modeling.

As association rules analysis is an important part of the proposed method, we study the features and characters of this methodology thoroughly. In practical terms, this technique is appropriate for selecting interactions among variables. The next step is to improve this methodology by selecting the most appropriate rules selection criteria. However, there is an issue
that, if being ignored, would cause an inappropriate comparison among the rules selection criteria. It is the circumstance called ‘equivalent models’, under which different models of rules could have generated identical responses. Our analysis shows that there can be myriads of models that are equally well suited to explaining the same dataset. The rules selection method needs to take this issue into serious consideration, as it plays a direct role in rules justification. Some rules may not contribute to some models but may make a major contribution to other models. Establishing the importance of each rule individually is inappropriate for an analysis when there are multiple ways to represent the data. In other words, if the issue of equivalent models is not understood correctly such that only the model of rules is considered, comparisons among many methods and criteria can be misleading. When a given methodology cannot capture the underlined model but captures equivalent models instead, designating this methodology as an incapable method would be incorrect. To our knowledge, the literature to date has not touched upon this major issue.

This issue of equivalent models and related subjects is the focus of the second part of our study. The objective is to explore the issue of the equivalent problem in a thorough way. We discuss all the essential ideas and analysis including the definitions, generation, verification, and theoretical properties of equivalent models. We illustrate the importance of this issue for association rules analysis and show how easily it can occur for any dataset. We focus on preventing misleading results and provide a solution for unstable explanations from equivalent models. Having considered the issues associated with equivalent models, we focus next on improving the features of association rules analysis through rules selection criteria for the proposed method.

The third part of this study considers how to improve the performance of the proposed method by applying rules selection criteria that are practical and robust for a variety of models in our scenarios of interest, which include (higher-order) interactions as parts of the models. As introduced in the early 1990s, numerous research studies have aimed to improve association rules analysis by applying alternative rules selection criteria. From a thorough search of the data mining papers and journals in the recent literature, almost 50 rules selection criteria have been applied or developed (Tan et al. 2004; Geng et al. 2006; Blanchard et al. 2009).

In this section, our objectives are to recommend criteria that are robust in selecting true models for a variety of scenarios and to analyze and compare the properties for criteria according
to their performances. Accordingly, we have developed a methodological framework for rules selection comparisons to provide fair comparisons and thorough analysis. In addition, the framework has been designed systematically in order to generate circumstances that comprehensively represent the scenarios of interest. We analyze and compare the performances of the rules selection criteria in capturing true models via a methodical simulation process. The advantages of our comparison framework are as follows. First, we provide the basis for comparing the rules selection criteria ability’s to search for all the possible true models rather than limiting the search to particular models or specific rules. Second, the performance comparison is based not results from specific datasets but on the simulation results from general datasets generated from the scenarios of interest. Third, we do not depend on the judgments of experts in this study. Fourth, addressing the issue of equivalent models is taken into serious consideration in this study as it plays an important part in the analysis and comparison. And, we propose this comparison framework as a benchmark for comparison procedures in future studies. Throughout the methodical simulation processes, the performance comparisons and recommendations for rule selection criteria are provided.

In the last section of our research, we will show that the compatibility of the proposed method is not limited to the logistic regression model but can be extended to the categorical model in general. The ability to search for the optimal models for other categorical models with more levels of response is also investigated. We demonstrate the applicability of our method to the multinomial logit model, which is an example of a more complicated model. Thus, the effectiveness and efficiency of our method when interactions are present is confirmed.

This research is motivated by data mining technique, specifically market basket analysis. We present an approach that has never been tried before: specifically, we apply association rules analysis in order to incorporate interactions into logistic regression modeling. The issue of equivalent models which we discover in our study is shown to have a major impact on association rules analysis such that comparisons and analyses of any techniques or rules selection criteria that do not account for this issue are likely to be misleading and invalid. We present a comprehensive study of the rules selection criteria that are most practical and robust for our proposed method. We extend the applications to multinomial logistic regression and confirm that the proposed method is not only practical for use with the logistic regression model but with categorical models in general.
Accordingly, this dissertation is organized as follows. Section 2 presents the model selection procedure for the logistic regression model via association rules analysis. Section 3 presents the issue of equivalent models in a thorough way. Section 4 presents the analysis of the rules selection criteria and comparison. Section 5 illustrates how the proposed method can be applied with a multinomial logit model. Section 6 offers a discussion, concluding remarks, and recommendations for future work.
Chapter 2

Logistic Regression via Association Rules Analysis

In this chapter, we introduce the logistic model selection procedure via association rules analysis. The collaboration between the two techniques constitutes the principal focus of this research. The most significant feature of association rules analysis is that it overcomes the disadvantage of the logistic regression model by accounting for interactions between the categorical variables. This chapter is organized as follows. Section 2.1 reviews the relevant academic literature pertaining to logistic regression modeling and association rules analysis. Section 2.2 presents the framework and discusses the proposed method in detail. Section 2.3 illustrates an example of our framework using a simulated dataset. Section 2.4 presents the application of our framework to a real dataset. Section 2.5 offers a discussion and concluding remarks.

2.1 Logistic Regression and Association Rules Analysis

Logistic regression has been widely used in many academic areas and applications. It has been used to explore relationships among variables and to predict response classes based on explanatory variable values. It was initially used in biomedical studies (Agresti, 2002); however, in recent decades it has been widely used in a number of other areas. In business, it has been used in applications involving credit scoring (Patra et al. 2008) and business forecasting and assessment (Lawson and Montgomery 2007; Lin et al. 2007). Other application areas include genetics (Coffey et al. 2004; Park and Hassie 2008), education (Morrell and Auer 2007), agriculture (Miramontes et al. 2006), engineering (Loh 2006; Kaur and Pulugurta 2008), neuroscience (Lu et al. 2004), and forestry (Yang et al. 2006; Yang et al. 2006).

A logistic regression model explains the functional relationship between a link function and predictors in a linear form. The difference between the linear model and the logistic regression model is that whereas the linear model uses an identity link function, the logistic regression uses a logit link function. The logistic regression model uses a combination of
categorical responses and explanatory variables, the latter of which can be quantitative variables, categorical variables, or both. The logistic regression model with only main effects is illustrated here:

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k$$

where \( p \) is the probability of success, \( X \)'s are the main effects, and \( \beta \)'s are the coefficients of these main effects.

Interactions have been addressed in different ways, especially interactions between categorical variables. According to Agresti (2002), interactions are typically omitted from classical logistic regression models. In other methods, interactions have been considered, but only after their main effects have been added to the model. Hosmer and Lemeshow (1989) considered interactions to be of interest after including main effects using the stepwise regression method. This type of analysis has two stages: the first stage considers only the main effects, and the second stage considers interactions among the significant main effects obtained in the first stage. Note that some applications allow models to add interactions without their main effects (Lin et al. 2008).

In our study, we consider interactions and main effects simultaneously in the logistic regression model. As the number of interactions increases at an accelerated rate as the number of main effects increases, the subset selection method is extremely inefficient when all interactions are considered at the same time. Therefore, an efficient method for selecting potential interaction variables is required. We propose to implement association rules analysis for this purpose.

Association rules analysis is a methodology for exploring relationships among items in the form of rules. Each rule has two parts. The first part includes left-hand side item(s), or condition(s), and the second part is a right-hand side item, or a result. The rule is always represented as a statement: if \( \text{condition} \) then \( \text{result} \) (Berry and Linoff 1997). Two measurements are attached to each rule. The first measurement is called support \((s)\), and it is computed by \( s = P(\text{condition and result}) \). The second measurement is called confidence \((c)\), and it is computed by \( c = \frac{P(\text{condition and result})}{P(\text{condition})} \). Association rules analysis finds all the rules that meet two key thresholds: minimum support and minimum confidence (Agrawal and Srikant 1994).
This technique is widely used in market basket analysis. Here, the items for association rules analysis are products (e.g., milk, bread, and cheese). In this case, the rule would be “If a person buys bread (denoted in the condition by A), then the person will buy milk (denoted in the result by B).” The rule can be rewritten as “if A then B.” The support and confidence for this rule are \( s = P(A \cap B) \) and \( c = P(A \cap B) / P(A) = P(B \mid A) \), respectively. On the other hand, if the rule is “if B then A,” the support and confidence are \( s = P(A \cap B) \) and \( c = P(A \cap B) / P(B) = P(A \mid B) \), respectively. The information from this set of rules can then be used for product management (e.g., deciding which products to display together). It is well known that when the rule “if A then B” has confidence \( c \), then the rule “if A then not B” will have confidence \( 1-c \). Thus, when \( c \) is very small, the rule “if A then B” will not be considered. The rule “if A then not B” should be considered, however.

This set of rules can be used for other purposes, including classification. A technique called classification rule mining (CRM), which is a subset of association rules analysis, was developed to find a set of rules in a database that would form an accurate classifier (Liu et al. 1998; Quinlan 1992). This technique uses an item to represent a pair consisting of a main effect and its corresponding integer value. More specific than association rules analysis, CRM has only one target, and this must be specified in advance. In general, the target of CRM is the response, which means the result of the rule (the right-hand side item) can only be the response and its class. Therefore, the left-hand side item (the condition) consists of the explanatory variable and its level. For example, assume there are \( k \) binary factors, \( X_1, X_2, \ldots, X_k \), and a binary response, \( Y \). All variables have two levels denoted by 0 and 1. Many rules can be generated by CRM. One of the rules could be “if \( X_1 = 1 \), then \( Y = 1 \)” with \( s = P(X_1 = 1 \text{ and } Y = 1) \) and \( c = P(X_1 = 1 \text{ and } Y = 1) / P(X_1 = 1) \). Another possibility is that the rule could be “if \( X_1 = 1 \), then \( Y = 0 \)” with \( s = P(X_1 = 1 \text{ and } Y = 0) \) and \( c = P(X_1 = 1 \text{ and } Y = 0) / P(X_1 = 1) \), among others.

To our knowledge, few studies have linked association rules analysis to logistic regression modeling. Freyberger et al. (2004) apply classification rules mining to aid in the selection of logistic regression terms in transfer models of student learning. They begin a sequential search algorithm by choosing sets of main effects with high confidence for response prediction to create the first sets of logistic regression models. Then, rule variables that help predict the response from the misclassification instances are added. The process is repeated until
adding new association rules no longer improves the misclassification rate. Cen et al. (2005) use a similar approach in their search for a cognitive model of student learning. In addition, Jaroszewicz (2006) applies association rules analysis in order to find polynomial terms to add to logistic regression.

Unlike previous work, our proposed method applies association rules analysis as part of the global search for the logistic regression model (Changpetch and Lin 2012). In our study, we apply classification rule mining in order to retain only the potentially significant variables for consideration in building the logistic regression model. This methodology is an important aspect of the variables selection in our process. The proposed method creates a rule conversion technique for transforming important rules into binary variables and performing a global search for the optimal model.

Several studies connect logistic regression and association rules analysis in support of achieving other objectives. For example, Shaharanee et al. (2009) apply logistic regression as a tool for finding significant rules in association rules analysis. Kamei et al. (2008) apply logistic regression with association rules analysis to create the fault-prone module prediction framework. The modules can be classified using a logistic regression model or association rules analysis, depending on the conditions of the modules themselves. Here, we are more interested in applying association rules analysis in order to find an optimal logistic model and to thereby facilitate understanding of datasets and improve model fit. There is a large body of research in the computer science literature that focuses on the different aspects, such as the accuracy of the classification method. However, our focus is on presenting our proposed method clearly; therefore, such issues will not be discussed further here.
2.2 The Proposed Method

The proposed framework for building a model that predicts a binary response from binary explanatory variables consists of four key steps (Changpetch and Lin 2012).

Figure 2-1: Framework for the proposed model building

As shown in Figure 2-1, the four steps in our framework are:

Step 1: Generate the rules from association rules analysis.

Step 2: Select the rules (based on confidence).

Step 3: Generate the variables for each rule from step 2.

Step 4: From variables in Step 3 and all the main effects, search for the optimal model.

Figure 2-2 displays the workflow of the proposed method. Details are discussed below.

**Step 1: Association Rules Analysis**

First, we use association rules analysis to create rules from datasets. Specifically, we perform CRM. For each rule, the condition (left-hand side items) represents the combination of the explanatory variables and their levels, whereas the result (right-hand side item) is the response class. To perform CRM, we use the CBA program developed by the Department of Information Systems and Computer Sciences at the National University of Singapore (Liu et al. 1998; website: http://www.comp.nus.edu.sg/~dm2/). With this program, we obtain all the active rules using the given minimum support and minimum confidence. Typically, we set the minimum support to 10% and the minimum confidence to 80%. However, we increase the minimum support to 15% if more than 1,000 rules are generated. The expected results from this
Step 1: Rule Generation

The rules are of the form “if \( X_i = x_i \) then \( Y = y \),” where \( x_i \) is the level of variable \( X_i \), and \( y \) is the level of response \( Y \). With each rule, the respective support and confidence are attached. All active rules become input variables for the second step.

**Step 2: Rule Selection**

Secondly, we select those top rules to convert into interaction variables in the next step. The rules with the highest confidence are selected from the first step. We call the rules selected at this stage **potential rules**. Note that the number of rules selected at this stage is relatively small compared to the total number of possible interactions for the dataset. As a rule of thumb, we set the number of potential rules at 30. This can be adjusted based on any prior knowledge of the problem. The more variables, the more potential rules should be chosen. In general, a total number of 30 to 50 rules is recommended. All potential rules are inputs for the third step.

**Step 3: Variable Generation**

In this step, we create the variables for logistic regression from the potential rules. To convert a rule into an interaction, we create interactions among the main effects on the left-hand side with the same settings that appeared in the rule. Suppose that the selected rule is “if \( X_i = x_i \) and \( X_j = x_j \) then \( Y = y \),” where \( x_i \) is the level of variable \( X_i \), \( x_j \) is the level of variable \( X_j \), and \( y \) is the level of response \( Y \). We generate an interaction between \( X_i \) and \( X_j \) by labeling this interaction as 1 if \( X_i = x_i \) and \( X_j = x_j \), and as 0 otherwise. This interaction is denoted by \( X_i(x_i)X_j(x_j) \). For example, for the rule if \( X_1 = 0 \) and \( X_2 = 1 \), then \( Y = 0 \), we create an interaction between \( X_1 \) and \( X_2 \) denoted by \( X_1(0)X_2(1) \). We have \( X_1(0)X_2(1) = 1 \) if \( X_1 = 0 \) and \( X_2 = 1 \), and 0 otherwise. Note that the level of \( Y \) does not play any role in variable generation. The results from this step, which are the binary variables generated from the potential rules, are inputs for the fourth step.

**Step 4: Model Selection**

In principle, any model selection criterion can be used. Here, the AIC (Akaike information criterion) is used for illustration (Akaike 1974). The selected proposed model is the one that gives the minimum AIC among all models. We describe the process in great detail. First, a set of candidate models must be selected. For each number of variables, the model with
the optimal log-likelihood value is selected and becomes a candidate model. All possible combinations are tested. The variables in this step consist of all the interactions generated in Step 3 and all the main effects. The optimal model with the lowest AIC among the candidate models is then selected (Agresti, 2002). Note that there are many choices in this model selection step. Other advanced model selection methodologies, e.g., lasso (least absolute shrinkage and selection operator; Tibshirani 1996), SCAD (smoothly clipped absolute deviation; Fan 1997), adaptive lasso; Zou 2006), LARS (least-angle regression algorithm; Efron et al. 2004), Dantzig selector (Candes and Tao 2007), or another selection criteria, e.g., BIC can be applied here.
Figure 2-2: Workflow of the proposed method
2.3 Illustrated Example Using the Proposed Method

For illustration and comparison, we use the MONK’s dataset (the first dataset from the MONK’s problem). It was compiled by Sebastian Thrun of Carnegie Mellon University using propositional formulas over six factors (Thrun et al. 1991). We obtained this dataset from the UCI machine learning website (http://archive.ics.uci.edu/ml/datasets.html). According to the website, the MONK’s problem is the basis for the first international comparison of learning algorithms. This dataset is constructed in such a way that there are interactions among variables.

The objective of this dataset is to classify 432 robots into two classes (class 0 and class 1) based on six attributes. The details for all the attributes are shown in Table 1. The true model would assign the robot into two classes with a subset of the six attributes. However, assuming that the true model is unknown, many studies use their own methodologies to predict the class of the robot. In this study, we use the proposed method to find the logistic regression that fits this dataset and then compare the results to those generated by the classical logistic regression technique.

Note that the true model for the MONK’s dataset is designed according to the following logic (the truth):

- If the head shape and the body shape of the robot are the same or the robot wears a red jacket, then the robot is in class 1.
- If the head shape and the body shape of the robot are not the same and the robot does not wear a red jacket, then the robot is in class 0.

Table 2-1: Attributes for MONK’s dataset

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Levels</th>
<th>Binary Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>head_shape</td>
<td>round, square, octagon</td>
<td>X₁ = 1 if head_shape is round and X₁ = 0, otherwise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X₂ = 1 if head_shape is square and X₂ = 0, otherwise</td>
</tr>
<tr>
<td>body_shape</td>
<td>round, square, octagon</td>
<td>X₃ = 1 if body_shape is round and X₃ = 0, otherwise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X₄ = 1 if body_shape is square and X₄ = 0, otherwise</td>
</tr>
<tr>
<td>is_smiling</td>
<td>Yes, no</td>
<td>X₅ = 1 if smiling and X₅ = 0, otherwise</td>
</tr>
<tr>
<td>holding</td>
<td>sword, balloon, flag</td>
<td>X₆ = 1 if holding a sword and X₆ = 0, otherwise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X₇ = 1 if holding a balloon and X₇ = 0, otherwise</td>
</tr>
<tr>
<td>jacket_color</td>
<td>Red, yellow, green, blue</td>
<td>X₈ = 1 if jacket color is red and X₈ = 0, otherwise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X₉ = 1 if jacket color is yellow and X₉ = 0, otherwise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X₁₀ = 1 if jacket color is green and X₁₀ = 0, otherwise</td>
</tr>
<tr>
<td>has_tie</td>
<td>Yes, no</td>
<td>X₁₁ = 1 if wearing a tie and X₁₁ = 0, otherwise</td>
</tr>
</tbody>
</table>
Table 2-1 presents details about the original attributes and their levels. All are multi-level categorical variables. To construct the logistic regression, we convert the attributes into binary variables (i.e., $X_1$, $X_2$, ..., $X_{11}$), as shown in Table 1. According to the binary variables and the true model, the robot will be in class 1 ($Y = 1$) or class 0 ($Y = 0$), based on the conditions shown in Table 2-2. This is the underlying (true) model. Applying the proposed method, we obtain the following results:

**Table 2-2: Variable class results for MONK’s dataset**

<table>
<thead>
<tr>
<th>Class 1 ($Y = 1$)</th>
<th>Class 0 ($Y = 0$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1 = 1$ and $X_3 = 1$ (the robot has a round head shape and a round body shape)</td>
<td>$X_1 = 1$, $X_3 = 0$ and $X_8 = 0$ (the robot (i) has a round head shape, (ii) does not have a round body shape, and (iii) does not wear a red jacket)</td>
</tr>
<tr>
<td>$X_2 = 1$ and $X_4 = 1$ (the robot has a square head shape and a square body shape)</td>
<td>$X_1 = 0$, $X_3 = 1$ and $X_8 = 0$ (the robot (i) does not have a round head shape, (ii) has a round body shape, and (iii) does not wear a red jacket)</td>
</tr>
<tr>
<td>$X_1 = 0$, $X_2 = 0$, $X_3 = 0$, and $X_4 = 0$ (the robot has an octagonal head shape and an octagonal body shape)</td>
<td>$X_2 = 1$, $X_4 = 0$ and $X_8 = 0$ (the robot (i) has a square head shape, (ii) does not have a square body shape, and (iii) does not wear a red jacket)</td>
</tr>
<tr>
<td>$X_8 = 1$ (the robot wears a red jacket)</td>
<td>$X_2 = 0$, $X_4 = 1$ and $X_8 = 0$ (the robot (i) does not have a square head shape, (ii) has a square body shape, and (iii) does not wear a red jacket)</td>
</tr>
</tbody>
</table>

**Step 1:** Use CBA to obtain the active rules. Note that we use a minimum support value of 10% and a minimum confidence value of 80% to generate the active rules, as discussed previously.

**Step 2:** Select the 30 rules with the highest confidence values from among all the active rules. These rules will be inputs for step 3. The top five rules are displayed below as examples.

Rule 1: If $X_8 = 1$, then $Y = 1$ with $s = 0.25$, and $c = 1.000$.
Rule 2: If $X_1 = 1$, $X_3 = 0$, $X_8 = 0$, then $Y = 0$ with $s = 0.1667$, and $c = 1.000$.
Rule 3: If $X_1 = 0$, $X_3 = 1$, $X_8 = 0$, then $Y = 0$ with $s = 0.1667$, and $c = 1.000$.
Rule 4: If $X_2 = 1$, $X_4 = 0$, $X_8 = 0$, then $Y = 0$ with $s = 0.1667$, and $c = 1.000$.
Rule 5: If $X_2 = 0$, $X_4 = 1$, $X_8 = 0$, then $Y = 0$ with $s = 0.1667$, and $c = 1.000$. 

15
Step 3: The 30 selected rules are converted into variables. Take rule 2 as an example. From this rule, as $X_1$, $X_3$, and $X_8$ contribute to $Y$ in a specific pattern, we create the new variable called $X_1(1)X_3(0)X_8(0)$, where

$$X_1(1)X_3(0)X_8(0) = \begin{cases} 1 & \text{if } X_1 = 1, X_3 = 0, X_8 = 0 \\ 0 & \text{otherwise} \end{cases}.$$

For the other example, rule 3 is converted into the new variable called $X_1(0)X_3(1)X_8(0)$, where

$$X_1(0)X_3(1)X_8(0) = \begin{cases} 1 & \text{if } X_1 = 0, X_3 = 1, X_8 = 0 \\ 0 & \text{otherwise} \end{cases}.$$

We generate all new variables in the same way unless the new variables duplicate input variables. Note that rule 1 generates variable $X_8$, which duplicates the input variable. Therefore, the total number of new generated variables is 29 instead of 30.

Step 4: The subset selection results are shown in Table 2-3. Note that the candidate variables are the main effects ($X_1$, $X_2$, ..., $X_{11}$) and all the potential interactions from step 3. Therefore, the total number of variables to be considered in this step is 40 (11 input variables and 29 new generated variables).

<table>
<thead>
<tr>
<th>k</th>
<th>Variables</th>
<th>log-likelihood</th>
<th>AIC</th>
<th>BIC</th>
<th>$R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$X_8$</td>
<td>-206.2602</td>
<td>416.5204</td>
<td>424.6573</td>
<td>0.3112</td>
</tr>
<tr>
<td>2</td>
<td>$X_1(0)X_3(1)X_8(0)$, $X_2(0)X_4(1)X_8(0)$</td>
<td>-161.9920</td>
<td>329.9840</td>
<td>342.1893</td>
<td>0.4590</td>
</tr>
<tr>
<td>3</td>
<td>$X_1(1)X_3(0)X_8(0)$, $X_2(0)X_4(1)X_8(0)$, $X_1(0)X_3(0)X_8(0)$</td>
<td>-81.0241</td>
<td>170.0482</td>
<td>186.3219</td>
<td>0.7294</td>
</tr>
<tr>
<td>4</td>
<td>$X_8$, $X_1(1)X_3(1)$, $X_2(1)X_4(1)$, $X_1(0)X_3(0)X_8(0)$</td>
<td>0.0000</td>
<td>10.0000</td>
<td>30.3421</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

We use the best subset selection method with AIC measurement and come up with the four-variable model as shown below. Note that this model fits perfectly due to the perfect $R^2$ value. The model is

$$\ln\left( \frac{p}{1-p} \right) = -20.8893 + 41.2947X_8 + 41.5142X_1(1)X_3(1) + 41.5142X_2(1)X_4(1)$$
$$+ 41.5142X_1(0)X_2(0)X_3(0)X_4(0).$$

According to estimated coefficients from the four variables, the model can be explained using four conditions according to each variable:
i. If $X_8 = 1$, then $Y = 1$.

ii. If $X_1(1)X_3(1) = 1$, meaning $X_1 = 1$ and $X_3 = 1$, then $Y = 1$.

iii. If $X_2(1)X_4(1) = 1$, meaning $X_2 = 1$ and $X_4 = 1$, then $Y = 1$.

iv. If $X_1(0)X_2(0)X_3(0)X_4(0) = 1$, meaning $X_1 = 0$, $X_2 = 0$, $X_3 = 0$ and $X_4 = 0$, then $Y = 1$.

These four conditions are exactly the same conditions for $Y = 1$ as shown in Table 2-2. Therefore, this model can capture the whole logic behind the true model.

Classical logistic regression models with optimal AIC, BIC (Bayesian Information Criterion), and $R^2$ values must be found in order to compare the results of our proposed model. To do this, we find candidate models for classical logistic regression that give the optimal log-likelihoods for each number of parameters, as in the fourth step of our proposed method. Then, all measurements for the candidate models are calculated to obtain the models with optimal AIC, BIC, and $R^2$. Note that the candidate variables for the classical models are only main effects. As the results show in Table 2-4, adding more variables when $X_8$ is already included in the model does not improve the log-likelihood, as the parameters for the other variables are equal to zero.

Table 2-4: Classical model for MONK's dataset

<table>
<thead>
<tr>
<th>k</th>
<th>Variables</th>
<th>log-likelihood</th>
<th>AIC</th>
<th>BIC</th>
<th>$R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$X_8$</td>
<td>-206.2602</td>
<td>416.5204</td>
<td>424.6573</td>
<td>0.3112</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>$X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}$</td>
<td>-206.2602</td>
<td>416.5204</td>
<td>424.6573</td>
<td>0.3112</td>
</tr>
</tbody>
</table>

Therefore, the classical model is $\ln \left( \frac{P}{1-P} \right) = -0.693 + 21.896X_8$.

Based on this model, the classical method only captures the significance of the color of the jacket ($X_8$), and does not capture the significance of the head shape or the body shape of the robot ($X_1, X_2, X_3,$ and $X_4$). Moreover, the fit of the model is also poor compared to the proposed model according to the value of $R^2$. Additionally, the prediction ability of the classical model is far inferior to that of the proposed model, as shown in Table 2-5.
Table 2-5: Comparison between the proposed model and the classical model for MONK’s dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>AIC</th>
<th>BIC</th>
<th>$R^2$</th>
<th>Prediction error†</th>
<th>Corr($Y, \hat{Y}$)‡</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed model</td>
<td>10.0000</td>
<td>30.3421</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>Classical model</td>
<td>416.5204</td>
<td>424.6573</td>
<td>0.3112</td>
<td>0.2500</td>
<td>0.5772</td>
</tr>
</tbody>
</table>

† Prediction error = number of misclassified cases / total number of cases. The method of k-fold cross-validation can also be applied here.
‡ $\text{Corr}(Y, \hat{Y})$ represents the Spearman’s correlation between the real class ($Y$) and the predicted class ($\hat{Y}$). Note that for the two binary variables ($Y$ and $\hat{Y}$), Spearman’s correlation and Kendall’s Tau-b are the same. Also note that for AIC, BIC, and the prediction error, the lower values are better. Conversely, for $R^2$ and correlation, the higher values are better.

The MONK’s dataset adequately illustrates our proposed method and demonstrates its ability to capture the interactions between variables. Our model outperforms the classical models on all measurements in the example, including AIC, BIC, $R^2$, the prediction error, and the correlation between the real class and the predicted class. Any method that does not take high-order interactions (three-order interactions and four-order interactions) into consideration would not be able to obtain the true model for the MONK’s dataset. Note that the logic whereby the robot is in class 1 if the head shape and body shape are the same cannot be fully captured if the method considers only main effects and two-way interactions.

One may consider using all possible combinations of interactions in the standard model selection methods without implementing association rules analysis. However, our empirical study indicates that such an approach has never led to the true model. This is mainly due to the fact that there are too many candidate variables involved (e.g., for 11 variables, there are more than 500 interactions when we consider up to 4-way interactions).

2.3.1 Comparisons Based on Simulated Datasets

Next, we added noise to the dataset in order to determine whether the proposed method would still find the true model and explain the relationships among variables as it does for the original dataset with no noise. We conducted three simulations with three different noise levels (5%, 10% and 20%). Then, we compared the proposed models from each case to the classical models with optimal AIC, BIC, $R^2$, prediction error, and correlation values.
Similar to the original dataset, the four-variable model for each case of noise is still able to capture the whole logic behind the true model. In contrast, the classical method can explain only part of the model: the significance of $X_8$ to class allocation (details are available on request). The proposed models also significantly outperform the classical models for all levels of noise as shown in Table 2-6. However, as expected, the results show that all measurements get worse with increased levels of noise.

Table 2-6: Comparison between the proposed model and the classical models

<table>
<thead>
<tr>
<th>Noise added</th>
<th>Model</th>
<th>AIC</th>
<th>BIC</th>
<th>$R^2$</th>
<th>Prediction error</th>
<th>Corr($Y, \hat{Y}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%</td>
<td>Proposed model</td>
<td>181.6816</td>
<td>202.0237</td>
<td>0.7133</td>
<td>0.0440</td>
<td>0.9120</td>
</tr>
<tr>
<td></td>
<td>Classical models</td>
<td>449.2036</td>
<td>457.3405</td>
<td>0.2599</td>
<td>0.2662</td>
<td>0.5426</td>
</tr>
<tr>
<td>10%</td>
<td>Proposed model</td>
<td>313.1654</td>
<td>333.5075</td>
<td>0.4938</td>
<td>0.1019</td>
<td>0.7963</td>
</tr>
<tr>
<td></td>
<td>Classical models</td>
<td>510.2652</td>
<td>519.7309</td>
<td>0.1607</td>
<td>0.3102</td>
<td>0.4384</td>
</tr>
<tr>
<td>20%</td>
<td>Proposed model</td>
<td>428.7266</td>
<td>449.0687</td>
<td>0.3008</td>
<td>0.1759</td>
<td>0.6482</td>
</tr>
<tr>
<td></td>
<td>Classical models</td>
<td>543.8490</td>
<td>552.0705</td>
<td>0.1155</td>
<td>0.3565</td>
<td>0.3582</td>
</tr>
</tbody>
</table>

For each level of noise in Table 2-6, we found optimal values for the five measurements among all the classical models and compared them to the values of the five measurements in the proposed model. The comparison confirms that on all five criteria the proposed model outperforms every classical model.

The simulations illustrate the major advantage of the proposed method in including association rules analysis in the model selection process. As seen earlier, the potential rules, which are later converted into potential interactions, are selected by this methodology. With the converting methodology, these interactions become potential variables for the logistic regression model. The final model is chosen through the model selection method. With this four-step model selection procedure, the true models are chosen as illustrated through the MONK’s dataset. The simulation results show that the true model could always be found via the proposed method with the recommended thresholds and number of rules. We illustrate the implementation of the proposed method based on a real-life dataset in the next section.
2.4 Application: Lymphography Dataset

In this section, we use a lymphography dataset from the University Medical Centre at the Institute of Oncology in Ljubljana, Yugoslavia, which was graciously provided by M. Zwitter and M. Soklic. Here, we illustrate an application of our proposed method by selecting the model for this dataset using our framework.

The lymphography dataset (Polat and Güneş 2007) contains records of lymph disease diagnoses in four classes: normal, metastases, malign lymph, and fibrosis. Originally, there were 148 total instances: 2 normal diagnoses, 81 metastases diagnoses, 61 malign lymph diagnoses, and 4 fibrosis diagnoses. As there were so few normal and fibrosis diagnoses, we did not include them in the analysis. The modified dataset, therefore, comprised two classes and 142 instances. The goal was to determine the response classes for metastases and malign lymph. There were 18 numerically valued attributes, which we converted into binary variables, listed in Table 2-7.

Table 2-7: Responses and attributes for the lymphography dataset

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Levels</th>
<th>Binary Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagnosis class</td>
<td>malignant lymph, metastases</td>
<td>Y = 1 if the class is malign lymph</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y = 0 if the class is metastases</td>
</tr>
<tr>
<td>Lymphatic</td>
<td>arched, deformed, displaced</td>
<td>X1 = 1 if arched and X1 = 0, otherwise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X2 = 1 if deformed and X2 = 0, otherwise</td>
</tr>
<tr>
<td>Block of afferent</td>
<td>yes, no</td>
<td>X3 = 1 if yes and X3 = 0, otherwise</td>
</tr>
<tr>
<td>Block of lymph c</td>
<td>yes, no</td>
<td>X4 = 1 if yes and X4 = 0, otherwise</td>
</tr>
<tr>
<td>Block of lymph s</td>
<td>yes, no</td>
<td>X5 = 1 if yes and X5 = 0, otherwise</td>
</tr>
<tr>
<td>Bypass</td>
<td>yes, no</td>
<td>X6 = 1 if yes and X6 = 0, otherwise</td>
</tr>
<tr>
<td>Extravagates</td>
<td>yes, no</td>
<td>X7 = 1 if yes and X7 = 0, otherwise</td>
</tr>
<tr>
<td>Regeneration</td>
<td>yes, no</td>
<td>X8 = 1 if yes and X8 = 0, otherwise</td>
</tr>
<tr>
<td>Early uptake</td>
<td>yes, no</td>
<td>X9 = 1 if yes and X9 = 0, otherwise</td>
</tr>
<tr>
<td>Lymph node dimensions</td>
<td>1, 2</td>
<td>X10 = 1 if 2 and X10 = 0, otherwise</td>
</tr>
<tr>
<td>Lymph nodes enlarged</td>
<td>1, 2, 3, 4</td>
<td>X11 = 1 if 1 and X11 = 0, otherwise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X12 = 1 if 2 and X12 = 0, otherwise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X13 = 1 if 3 and X13 = 0, otherwise</td>
</tr>
<tr>
<td>Changes in lymph</td>
<td>bean, oval, round</td>
<td>X14 = 1 if bean and X14 = 0, otherwise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X15 = 1 if oval and X15 = 0, otherwise</td>
</tr>
<tr>
<td>Defects in node</td>
<td>no, lacunars, lacunars</td>
<td>X16 = 1 if no and X16 = 0, otherwise</td>
</tr>
<tr>
<td></td>
<td>marginal, lacunars central</td>
<td>X17 = 1 if lacunars and X17 = 0, otherwise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X18 = 1 if lacunars marginal and X18 = 0, otherwise</td>
</tr>
<tr>
<td>Changes in node</td>
<td>no, lacunars, lacunars</td>
<td>X19 = 1 if no and X19 = 0, otherwise</td>
</tr>
</tbody>
</table>
In the original dataset, there are more levels within some variables. However, when we deleted the six instances of normal and fibrosis diagnoses, some levels were eliminated. We decided to eliminate the attribute *change in structure*, as we did not have a solid criterion on which to discretize it because it had eight levels within the binary variable. We discretized *number of nodes* by dividing the observations into three groups based on distribution characteristics. We applied the proposed method to this dataset and obtained the following results:

**Step 1:** Used CBA to obtain the active rules. Note that we used a minimum support value of 15% and a minimum confidence value of 80% to generate the active rules. We increased the minimum support to 15%, as more than 1,000 rules were generated with 10% minimum support.

**Step 2:** Selected the 50 rules with the highest confidence values from among all the active rules. Note that we selected more rules than we did with the MONK’s dataset, as the present dataset contained many more binary variables. Examples of the selected rules follow:

Rule 1: If $X_3 = 0$, $X_9 = 1$, $X_{15} = 1$, $X_{22} = 0$ then $Y = 1$ with $s = 0.1761$, $c = 1.000$.

Rule 2: If $X_{21} = 0$, $X_{23} = 0$, $X_{26} = 0$ then $Y = 1$ with $s = 0.2958$, $c = 0.9762$.

Rule 3: If $X_9 = 1$, $X_{12} = 0$, $X_{21} = 0$, $X_{23} = 0$ then $Y = 1$ with $s = 0.2535$, $c = 0.9722$.

Rule 4: If $X_3 = 1$, $X_{12} = 1$, $X_{21} = 1$ then $Y = 0$ with $s = 0.2324$, $c = 0.9697$.

Rule 5: If $X_3 = 0$, $X_9 = 1$, $X_{18} = 0$, $X_{22} = 0$ then $Y = 1$ with $s = 0.2113$, $c = 0.9667$.

**Step 3:** The 50 selected rules were converted into variables. For example, rule 1 was converted into the new variable called $X_3(0)X_9(1)X_{15}(1)X_{22}(0)$, where
\[ X_3(0)X_9(1)X_{15}(1)X_{22}(0) = \begin{cases} 1 & \text{if } X_3 = 0, X_9 = 1, X_{15} = 1, X_{22} = 0 \\ 0 & \text{otherwise} \end{cases} \]

**Step 4:** The subset selection results are shown in Table 2-8. Note that the candidate variables were the main effects \((X_1, X_2, \ldots, X_{27})\) and all the potential interactions from step 3.

**Table 2-8: Candidate models for the lymphography dataset**

<table>
<thead>
<tr>
<th>(k)</th>
<th>Variables</th>
<th>log-likelihood</th>
<th>AIC</th>
<th>BIC</th>
<th>(R^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(X_{21}(0)X_{23}(0)X_{26}(0))</td>
<td>-54.7664</td>
<td>113.5328</td>
<td>119.4445</td>
<td>0.4355</td>
</tr>
<tr>
<td>2</td>
<td>(X_{21}(0)X_{23}(0)X_{26}(0),) (X_6(0)X_4(1)X_{15}(1)X_{22}(0))</td>
<td>-33.4935</td>
<td>72.9870</td>
<td>81.8545</td>
<td>0.6548</td>
</tr>
<tr>
<td>3</td>
<td>(X_{21}(0)X_{23}(0)X_{26}(0),) (X_3(0)X_6(1)X_{15}(1)X_{22}(0),) (X_3(0)X_{11}(0)X_{22}(0)X_{33}(1))</td>
<td>-22.7467</td>
<td>53.4934</td>
<td>65.3167</td>
<td>0.7655</td>
</tr>
<tr>
<td>4</td>
<td>(X_{21}(0)X_{23}(0)X_{26}(0),) (X_3(0)X_6(1)X_{15}(1)X_{22}(0),) (X_3(0)X_{11}(0)X_{22}(0)X_{33}(1),) (X_4(0)X_{23}(0)X_{26}(0)X_{27}(0))</td>
<td>-17.4160</td>
<td>44.8320</td>
<td>59.6111</td>
<td>0.8205</td>
</tr>
<tr>
<td>5</td>
<td>(X_7, X_{21}(0)X_{23}(0)X_{26}(0),) (X_3(0)X_6(1)X_{15}(1)X_{22}(0),) (X_3(0)X_{11}(0)X_{22}(0)X_{25}(1),) (X_4(0)X_{23}(0)X_{26}(0)X_{27}(0))</td>
<td>-12.4390</td>
<td>36.8780</td>
<td>54.6130</td>
<td>0.8718</td>
</tr>
</tbody>
</table>

We selected the five-variable model with the best AIC to represent this dataset. The model is

\[
\ln \left( \frac{p}{1 - p} \right) = -34.6770 + 31.7325X_7 + 35.7756X_{21}(0)X_{23}(0)X_{26}(0) + 52.4120X_3(0)X_9(1)X_{15}(1)X_{22}(0) \\
+ 35.7756X_3(0)X_{11}(0)X_{22}(0)X_{24}(1) + 19.1184X_6(0)X_{23}(0)X_{26}(0)X_{27}(0).
\]

According to the estimated coefficients from the five variables, the model can be explained using five conditions for each variable:

i. If *extravagates* is yes, then the probability that the patient is in the malign lymph class will increase.

ii. If *changes in node* is not lacunars marginal, *special forms* is not chalices, and *number of nodes* is not 0–9, then the patient is predicted to be in the malign lymph class.

iii. If *block of afferent* is no, *early uptakes* is yes, *changes in lymph* is oval, and *special forms* is either chalices or vesicles, then the patient is predicted to be in the malign lymph class.
iv. If the block of afferent is no, lymph nodes enlarged is not 1, special forms is either chalices or vesicles, and dislocation of node is yes, then the patient is predicted to be in the malign lymph class.

v. If bypass is no, special forms is not chalices, number of nodes is not 0–9, and number of nodes is not 10–19, then the probability that the patient is in the malign lymph class will increase.

One reason why we stopped the search process at five variables is because the five-variable model outperforms the classical models on all criteria, as shown in Table 2-9.

**Table 2-9: Comparison between the proposed model and the classical models (lymphography)**

<table>
<thead>
<tr>
<th>Model</th>
<th>AIC</th>
<th>BIC</th>
<th>R²</th>
<th>Prediction error</th>
<th>Corr((Y, \hat{Y}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed model</td>
<td>36.8780</td>
<td>54.6130</td>
<td>0.8718</td>
<td>0.0282</td>
<td>0.9425</td>
</tr>
<tr>
<td>Classical models</td>
<td>80.2224</td>
<td>109.6556</td>
<td>0.7682</td>
<td>0.0493</td>
<td>0.8997</td>
</tr>
</tbody>
</table>

As shown in Table 2-9, we compared the proposed model (the five-variable model) with the classical models that give the optimal values for each criterion. The comparison shows that on all five criteria the proposed model outperformed all the classical models. Therefore, the proposed method better explains the lymphography dataset and improves predictions of lymph disease.

### 2.5 Discussion and Conclusion

A central problem in logistic regression modeling is that of accounting for interaction effects, specifically those between categorical variables. In this study, we propose a framework that is capable of selecting potential interactions from a large number of candidates and including those selections in the variable selection process. Typically neglected in logistic model building, significant higher-order interactions can be selected and incorporated into the model. Our study confirms that the methodology proposed herein is effective in selecting interactions that improve model fit and facilitate our understanding of datasets. The popular model selection methods in logistic regression do not work well in general, as indicated in the simulation.
There is some arbitrariness in the proposed method, e.g., the minimum support in step 1 and the number of selected rules in step 2. Determining the cut-off points has always been an issue for association rules analysis. The minimum thresholds always depend on practitioners. We propose a model-building procedure based on the commonly used threshold, which also works well with our dataset. The simulations also confirm that the recommended threshold works well with general datasets with the involvement of interactions.

If validation is of interest, splitting an entire dataset for cross validation could also be considered, although this paper does not include this step. Note that association rules are not the only interaction effects that can be used in this model. Other methods of discovering interaction effects can be employed here as well, as long as they are capable of detecting (low-order and high-order) interaction effects in an efficient way.
Chapter 3

Identifiability for Association Rules Analysis

Since it was introduced, association rules analysis has been refined and modified in many ways. A number of studies have develop methods for rules selection whereby some rules are preferred over others based on their criteria and whereby still other rules are considered irrelevant and are, therefore, left out of consideration (Liu et al. 1998; Dong et al. 1999; Li et al. 2001; Webb and Zhang 2005; Yin and Han 2003; Wan et al. 2009; Li 2006).

However, there is an issue that if ignored would cause an inappropriate comparison among the rules selection methodologies and criteria to occur. This is the circumstance referred to as “equivalent models,” under which different models of rules could generate identical responses. Our analysis shows that numerous models can be equally well suited to explaining the same dataset. The rules selection method needs to take this issue into serious consideration, as it plays a direct role in rules justification. Some rules may not contribute to some models but may make a major contribution to other models. Establishing the importance of each rule individually is inappropriate for an analysis when there are multiple ways to represent the data. In other words, if the issue of equivalent models is not understood correctly such that only one model of rules is considered, comparisons among many methods and criteria can be misleading. When a given methodology cannot capture the underlined model but captures equivalent models instead, designating this methodology as an incapable method would be incorrect. To our knowledge, the literature to date has not touched on this major issue.

Another major issue is that of explaining the relationship between input variables and responses when a large number of models represent the same dataset. On this point, we find that for all the equivalent models, there is one unique model to which they all refer called the “dominant model.” We propose the dominant model as a benchmark for explaining the relationship between variables for all the equivalent models. With this model, the explanation of the relationships among the variables will be stable and concise (Changpetch and Lin 2012).

The main goal of this chapter is to explore the issue of the equivalent problem in a thorough way. We discuss all the essential ideas and analysis including the definitions,
generation, verification, and theoretical properties of equivalent models. We illustrate the importance of this issue for association rules analysis and show how easily it can occur for any dataset. Overall, we focus on preventing misleading results and provide a solution for unstable explanations from equivalent models.

This chapter is organized as follows. Section 3.1 discusses the basic concept of equivalent models through an example. Section 3.2 presents all the relevant definitions, the generation, and the verification of the equivalent models. Section 3.3 provides the theoretical properties of the equivalent models. Section 3.4 describes the procedure for obtaining the dominant model. Section 3.5 illustrates how this problem is related to the well-known MONK’s dataset. Section 3.6 offers a discussion and concluding remarks.

### 3.1 An illustrative example

We first illustrate the identifiability problem of association rules analysis through an example. The example here shows how multiple models can be used to explain the same dataset. Note that this problem is very common and can exist in any dataset.

Consider the example of a symmetric design with five binary variables ($X_1$–$X_5$) and the binary response ($Y$) with classes 0 and 1. We use the model of rules, referred to as Model A, to generate the response $Y$. Model A is combined with two rules (Rule A1 and Rule A2) as follows.

- **Rule A1**: If $X_1 = 1$ then $Y = 1$
- **Rule A2**: If $X_2 = 1$ and $X_3 = 1$ then $Y = 1$

The dataset generated from this model is given in Table 3-1.

<table>
<thead>
<tr>
<th>Instance</th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
<th>$X_5$</th>
<th>Response ($Y$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3-1: Dataset for Model A
The generated dataset can be used to compare the respective performances of each methodology. The common comparison technique relies on observing whether the methods of interest discover the underlined rules. For this dataset, any technique that finds both Rule A1 and Rule A2 is considered competent.

Now consider the following model of rules, called Model B.

Rule B1: If $X_1 = 1$ and $X_2 = 1$ then $Y = 1$
Rule B2: If $X_1 = 1$ and $X_3 = 1$ then $Y = 1$
Rule B3: If $X_1 = 1$, $X_2 = 0$, and $X_3 = 0$ then $Y = 1$
Rule B4: If $X_1 = 0$, $X_2 = 1$, and $X_3 = 1$ then $Y = 1$

If any given rules selection technique selects this model (Model B) to represent the dataset, would the technique be rightly considered incompetent as it does not find the given
model (Model A)? To do so would be erroneous, as Model B generates responses that are identical to those produced by Model A. This is a major issue that we explore explicitly in this research. The lack of knowledge about equivalence among models causes significant problems for attempts to develop rules selection techniques, especially when all the techniques are compared. If all the equivalent models are not considered, how can the comparisons be accurate? The concept of equivalency and methods for generating equivalent models are discussed in detail in Section 3.2. Note that all the models shown in Table 3-2 will generate results identical to those in Table 3-1.

**Table 3-2: Illustrated equivalent models**

<table>
<thead>
<tr>
<th>Model</th>
<th>Rules</th>
</tr>
</thead>
</table>
| C     | Rule C1: If $X_1 = 0$ and $X_2 = 0$ then $Y = 0$  
    Rule C2: If $X_1 = 0$ and $X_3 = 0$ then $Y = 0$ |
| D     | Rule D1: If $X_1 = 1$ then $Y = 1$  
    Rule D2: If $X_1 = 0$, $X_2 = 1$ and $X_3 = 1$ then $Y = 1$ |
| E     | Rule E1: If $X_1 = 0$ and $X_2 = 0$ then $Y = 0$  
    Rule E2: If $X_1 = 0$, $X_2 = 1$ and $X_3 = 0$ then $Y = 0$ |
| F     | Rule F1: If $X_1 = 1$ and $X_4 = 0$ then $Y = 1$  
    Rule F2: If $X_1 = 1$ and $X_4 = 1$ then $Y = 1$  
    Rule F3: If $X_2 = 1$ and $X_3 = 1$ then $Y = 1$ |

In association rules analysis, many rules are typically eliminated, as they are considered irrelevant. However, the above example shows that a lot more rules may make a significant contribution to the dataset when used in different combinations. The next section presents a thorough study of equivalent models including definitions and how to produce them. The verification for equivalency is also provided.
3.2 Definitions and Generation of Equivalent Models

This section presents a comprehensive discussion of the concept of equivalency. The section begins with definitions of all the relevant terms used in this research. Then, we present three standard ways to generate equivalent models in order to demonstrate that equivalency can easily be generated among the models. We finish by presenting a method for verifying model equivalency.

3.2.1 Definitions

(i) Models are called equivalent if they generate identical responses.

Note that Models A to F in Section 3.1 are all equivalent. For simplicity, our study separates the equivalent models into two classes, class 0 and class 1, based on class of response. The equivalent model with all the rules giving conditions for \( Y = 0 \) is called the equivalent model for class 0. Likewise, the equivalent model with all the rules giving conditions for \( Y = 1 \) is called the equivalent model for class 1. Note that the equivalent models for different classes can be seen as complementary, as will be discussed later.

(ii) The dominant model is the model for which the total number of condition terms is minimal among all the equivalent models for either class \( Y = 1 \) or \( Y = 0 \).

Note that a term is represented by one variable and its level, e.g., \( X_i = 1 \). Note, too, that we count all the repeated condition terms. As there are two classes for a binary response, there are two dominant models—one dominant model for each class. In addition, the dominant model for each class is unique. The proof is provided in Section 3.3.

Consider the example in Section 3.1: Model A is the dominant model for class 1, and Model C is the dominant model for class 0. For Model A, the total number of condition terms is three: one term from Rule A1 \((X_1 = 1)\) and two terms from Rule A2 \((X_2 = 1 \text{ and } X_3 = 1)\). This number is the lowest of all the equivalent models for class 1. For Model C, the total number of condition terms is four: two terms from Rule C1 \((X_1 = 0 \text{ and } X_2 = 0)\) and two terms from Rule C2 \((X_1 = 0 \text{ and } X_3 = 0)\). This number is the lowest of all the equivalent models for class 0.
(iii) **Dominant rules** are all the rules in the dominant models.

As shown in Table 3-2, Rules A1, A2, C1, and C2 are dominant rules.

(iv) A **Disjoint model** is a model wherein no rules have overlapping condition spaces. Models D and E are disjoint models, whereas Model A is not a disjoint model, as Rule A1 and Rule A2 do not overlap (for example, instances 1, 9, 17, and 25 in Table 1).

---

**Figure 3-1: Dominant Model vs. Disjoint Model**

There are specific relationships among all equivalent models. Next, we illustrate how a dominant model can be used to generate other equivalent models.

### 3.2.2 Generating Equivalent Models

Essentially, there are three ways to generate equivalent models from the dominant model. The three general methods for generating equivalent models are as follows.

**Method 1: Complement Method**

This method is used to create a model depicting a class that differs from the class of the given model. If the given model is the model for \( Y = 1 \), then the complement model will be the model for \( Y = 0 \). Note that any given model can generate a complement model. However, we use the dominant model to show how to obtain the complement model as follows.
Suppose the dominant model, referred to as **Model (1)**, can be represented as

\[ \text{If } \bigcup_{i=1}^{d} \bigcap_{j=1}^{k} m_{ij} \text{ then } Y = 1. \]

(1)

Specifically, it has \( \mathcal{d} \) rules:

- Rule 1: If \( M_1 \) then \( Y = 1 \) or if \( \left( m_{11} \cap m_{12} \cap \ldots \cap m_{1k_1} \right) \) then \( Y = 1 \);
- Rule 2: If \( M_2 \) then \( Y = 1 \) or if \( \left( m_{21} \cap m_{22} \cap \ldots \cap m_{1k_2} \right) \) then \( Y = 1 \);

... … ...

- Rule \( \mathcal{d} \): If \( M_d \) then \( Y = 1 \) or if \( \left( m_{d1} \cap m_{d2} \cap \ldots \cap m_{dk_d} \right) \) then \( Y = 1 \);

where \( M_1, M_2, \ldots, M_d \) are the conditions for rules 1, 2, …, \( \mathcal{d} \), respectively, and \( Y \) is the response (either 0 or 1, we will use 1, without loss of generality).

Note that \( m_1, m_2, \ldots, m_d \) are the terms of the inside conditions \( M_1, M_2, \ldots, M_d \), respectively, and \( k_1, k_2, \ldots, k_d \) are the numbers of the terms for \( M_1, M_2, \ldots, M_d \), respectively.

The complement model is then in the following form:

If \( \bigcup_{all \ j_1, j_2, \ldots, j_d} \left( m_{1j_1}^c \cap m_{2j_2}^c \cap \ldots \cap m_{dj_d}^c \right) \) then \( Y = 0 \), where \( j_1 = 1..k_1, j_2 = 1..k_2, \ldots, j_d = 1..k_d \).

For the model if \( (M_1 \cup M_2 \cup \ldots \cup M_d) \) then \( Y = 1 \), the complement model that yields \( Y = 0 \) is that if \( (M_1 \cup M_2 \cup \ldots \cup M_d)^c \) then \( Y = 0 \). With DeMorgan’s law, we can rewrite the complement model as if \( (M_1^c \cap M_2^c \cap \ldots \cap M_d^c) \) then \( Y = 0 \). The next step is to work with \( (M_1^c \cap M_2^c \cap \ldots \cap M_d^c) \).

Now, \( M_1^c = (m_{11} \cap m_{12} \cap \ldots \cap m_{1k_1})^c = m_{11}^c \cup m_{12}^c \cup \ldots \cup m_{1k_1}^c \);

\( M_2^c = m_{21}^c \cup m_{22}^c \cup \ldots \cup m_{2k_2}^c \); and \( M_d^c = m_{d1}^c \cup m_{d2}^c \cup \ldots \cup m_{dk_d}^c \).

Thus,

\[ \left( M_1^c \cap M_2^c \cap \ldots \cap M_d^c \right) = \left( \bigcup_{j_1=1}^{k_1} m_{1j_1}^c \right) \cap \left( \bigcup_{j_2=1}^{k_2} m_{2j_2}^c \right) \cap \ldots \cap \left( \bigcup_{j_d=1}^{k_d} m_{dj_d}^c \right) \]

\[ = \bigcup_{all \ j_1, j_2, \ldots, j_d} \left( m_{1j_1}^c \cap m_{2j_2}^c \cap \ldots \cap m_{dj_d}^c \right) \] as shown above.
Consider Model A from Section 3.1: Rule A1 is “if $X_1 = 1$ then $Y = 1$”; whereas Rule A2 is “if $X_2 = 1$ and $X_3 = 1$ then $Y = 1$.” Namely, $m_{11} = \{X_1 = 1\}$, $k_1 = 1$, $m_{21} = \{X_2 = 1\}$, $m_{22} = \{X_3 = 1\}$, and $k_2 = 2$.

For if $(X_1 = 1) \cup [(X_2 = 1) \cap (X_3 = 1)]$ then $Y = 1$, we have the complement model:

- if $\big((X_1 = 1)^c \cap [(X_2 = 1)^c \cup (X_3 = 1)^c]\big)$ then $Y = 0$;
- if $\big((X_1 = 0) \cap [(X_2 = 0) \cup (X_3 = 0)]\big)$ then $Y = 0$; and
- if $\big[(X_1 = 0) \cap (X_2 = 0)\big] \cup [(X_2 = 0) \cap (X_3 = 0)]$ then $Y = 0$.

We can write this complement model as Model C, as shown in Table 3-2.

**Method 2: Partition Method**

This method generates equivalent models by partitioning the dominant rules. Partitioning the rules is the same as adding more terms to the rules as shown below.

This type of equivalent model can be obtained by adding more term(s) and its complement to the condition(s) of the dominant rule. Therefore, one dominant rule will be expanded into multiple rules. Suppose the dominant rule is “if $M_1$ then $Y = 1$.” Adding more conditions (e.g., for condition $M^c_1$), the first rule is “if $M_1$ and $M^c_1$ then $Y = 1$” and the second rule is “if $M_1$ and $M^c_1$ then $Y = 1$.” The combination of the first rule and the second rule is equivalent to the original dominant rule.

As the example shown in Table 3-2 indicates, adding $X_4 = 0$ to Rule A1 will generate Rule F1. Similarly, adding $X_4 = 1$ to Rule A2 generates Rule F2. Model F, therefore, is equivalent to Model A.
As illustrated in Figure 3-2, this method is the same as partitioning the dominant rules into multiple rules. This method can generate numerous equivalent models, as dominant rules can be partitioned in multiple ways.

Method 3: Disjoint Method

This method generates an equivalent model with less overlapping condition spaces among the rules. If there is no overlapping space among the rules, then we have the disjoint model. This type of equivalent model is generated by adding the complement condition(s) of the dominant rule(s) to the other dominant rule(s) in a single given equivalent model.

Consider the dominant model Model (1): adding the complement condition $M_1^c$ to any of the other dominant rule(s) generates the other equivalent model. Note that $M_1^c$ can be added to multiple dominant rules at the same time, which generates different equivalent models; i.e., it is not limited to being added to just one other dominant rule. Consider the example in Table 3-2: the complement condition of Rule A1 is $X_1 = 0$. Adding this condition to Rule A2 generates Rule D2 in Model D. Likewise, we also have the option of adding the complement condition of Rule A2 to Rule A1 in order to generate the other equivalent model. With this method, the overlapping spaces among the conditions are reduced.

The disjoint equivalent model can be obtained by adding cumulative complement conditions to the dominant rules:

$$\text{If } [M_1 \cup (M_1^c \cap M_2) \cup \ldots \cup (M_1^c \cap M_2^c \cap \ldots \cap M_{d-1}^c \cap M_d)] \text{ then } Y = 1.$$ 

As the order of the dominant rules can be changed, multiple disjoint equivalent models can exist for each dataset. The graphical representation of this method is presented in Figure 3-3.

![Figure 3a](image1.png) ![Figure 3b](image2.png) ![Figure 3c](image3.png)

Figure 3-3: Generation of disjoint model
Figure 3a represents the dominant model with three rules. The equivalent model represented by Figure 3b is generated by adding the complement condition of rule d2 to both rules d1 and d3. The equivalent model represented by Figure 3c is generated by further adding the complement condition of rule d1 to rule d3 so that this is indeed a disjoint model.

3.2.3 Verification of Equivalency

One way to justify the equivalency among the models is to compare their responses. If they are all identical, i.e., if all the instances are identical, the models are equivalent. Suppose the response of the true model is $Y_t$ and the response of the model under study is $Y_s$, the two models are equivalent if $\sum_{i=1}^{n} (Y_{ti} - Y_{si})^2 = 0$, where $n$ is the total number of instances.

Note that we have studied several other ways to verify equivalency, e.g. by comparing their complement models and by deriving the disjoint models. Depending on the circumstances, other methods may be preferred, but these are more complicated than the suggested method and are not discussed here.
3.3 Theoretical Properties

In this section, we present some theoretical properties. These properties distinguish equivalent models from non-equivalent models. We also prove the theorem that the dominant model is unique. In the discussion presented next, a child rule is the rule whose condition is the subset of its parent rule. As an example, consider Model A and Model B in Section 3.1, wherein Rule A1 is the parent rule of Rules B1, B2, and B3, whereas Rule A2 is the parent rule of Rule B4. In the other words, Rules B1, B2, and B3 are the child rules of Rule A1, whereas Rule B4 is the child rule of Rule A2.

Property 1: Any rule in any equivalent model must be a child rule of dominant rules or a dominant rule itself.

By definition, the dominant rules define the scope of the conditions for the defined class. Including any rule that is not a child rule of the dominant rules will introduce instance(s) beyond the scope of the equivalent models.

Property 2: For any equivalent models, their complement models are also equivalent.

This property can be shown through graphical presentation (refer to Figure 3-3 for an example). The three models are equivalent, with the area inside the three circles belonging, for example, to class 1. The complement models for these three models will cover the same area (outside the three circles) for class 0. The three models and their complement models are clearly equivalent in generating responses.

Theorem 1: The dominant model for each class is unique.

Proof: Given that the dominant model is Model (1), the complement model of this dominant model is in the form

\[
\text{If } \bigcup_{j_1, \ldots, j_d} (m_{j_1}^c \cap m_{j_2}^c \cap \ldots \cap m_{j_d}^c) \text{ then } Y = 0. \tag{a}
\]

From property 1, the rule in the equivalent models has to be either the child rule of the dominant rule or the dominant rule itself. Therefore, the complement model of any equivalent model will be in the form
If \[ \bigcup_{\text{all } j_1, j_2, \ldots, j_d} (m_{ij_1}^c \cap m_{ij_2}^c \cap \ldots \cap m_{ij_d}^c) \bigcup \text{ extra terms} \] then \( Y = 0 \), (b)

where extra terms are the products of excessive condition terms.

From property 2, the complement models of the equivalent models will be equivalent. This implies that Model (a) and Model (b) are equal. Thus, the extra terms, \( \bigcup \text{ extra terms} \), are redundant and can be eliminated.

Note that we can further compress the term \[ \bigcup_{\text{all } j_1, j_2, \ldots, j_d} (m_{ij_1}^c \cap m_{ij_2}^c \cap \ldots \cap m_{ij_d}^c) \] by eliminating any redundant terms or irrelevant rules to establish a dominant model for class 0. The term \[ \bigcup_{\text{all } j_1, j_2, \ldots, j_d} (m_{ij_1}^c \cap m_{ij_2}^c \cap \ldots \cap m_{ij_d}^c) \] is unique; therefore, the dominant model is unique. This completes the proof for the theory.

**Lemma**: all the condition terms in the dominant model will appear in all the equivalent models.

The importance of uniqueness suggests that all the equivalent models can refer to this dominant model. The proof implies that all the equivalent models can be compressed into the simplest final form, i.e., the dominant model. We propose to use this model as a benchmark for explaining the relationships among variables for all the equivalent models. The procedure for obtaining the dominant model is provided in the next section.
3.4 Representative for the Equivalent Models

Here, we propose the dominant model as a representative of all the equivalent models, as this model contains the most concise form among all the equivalent models. The methodology for obtaining the dominant model is presented next.

Procedure for Obtaining the Dominant Model

As previously discussed, all the sets of condition terms in the dominant model will appear in all the equivalent models. The dominant model can be obtained by deleting all the irrelevant terms. We propose the iterative method for eliminating irrelevant rules and irrelevant terms as follows (also refer to Figure 3-4).

Step 1: Eliminate irrelevant rule

Any rule that does not contribute to finding identical responses is excluded. Suppose there are k rules at the beginning. We search for rules that can be omitted while keeping the identical response under the remaining k-1 rules. If such a rule is found, it will be eliminated. Whether or not such an irrelevant rule is found, we proceed to step 2.

Step 2: Eliminate irrelevant term

From the model obtained in step 1, search for any condition term that can be omitted and retain the identical response. Omit only one term from one rule each time. If the selected term appears in multiple rules, omit it for only one rule.

- If a term that can be omitted, delete it from the corresponding rule, and repeat steps 1 and 2.
- If no terms can be omitted, this also implies that no irrelevant rules exist. Thus, the dominant model is found.
In Figure 3-4, there are two decisions to make. The first decision is to justify the irrelevant rule and the second decision is to justify the irrelevant term. The process will stop when there is no irrelevant term in the model. The final form after deleting all irrelevant rules and irrelevant terms is the dominant model.
3.5 Revisit MONK's dataset

Equivalent models can cause problems. The dataset we use here as an example is the MONK’s dataset (the first dataset from the MONK’s problem). We also used this dataset in Section 2.3. Numerous studies have used this dataset to compare methodologies. Among these, some of the papers that focus on finding the classification error rate without mentioning the model are Philip and Joseph (2000), Karacali et al. (2004), Hassen and Tazaki (2005), Shin (2006), Naval Jr. and Yusiong (2007), Zang and Wang (2008), Appavu and Rajaram (2008), and Qodmanan et al. (2010). Additionally, other studies search for models that represent the dataset (Marghny and El-Semman 2005; Etchells and Lisbao 2006; Guerreiro and Trigueiros 2010). These papers focus on the underlined rules that are given to the dataset without considering other possible equivalent models. If such methodologies discover the given rules, these studies consider them to be competent. The problem, as we demonstrate in Section 3.1, is that the methodology can be competent even when the underlined rules are not found as other equivalent models can be found instead. Note that the original paper (Thrun et al., 1991) does not refer to the possibility of generating other models either.

In the literature, many criteria have been proposed to evaluate the goodness of important rules and comparisons among models. We show that equivalent models can explain the relationships between input variables and responses in different ways. Without an understanding of the concept of equivalent models, the selection criteria can be misleading, and the comparison can be biased. Moreover, we suggest using the dominant model as a benchmark for a solid and concise explanation. The derivation used to obtain the dominant model is also presented.

Section 3.5.1 demonstrates equivalent models for this dataset. Section 3.5.2 describes the procedure for obtaining the dominant model.
3.5.1 Equivalent Models for MONK’s dataset

On the basis of the given logic with binary variables, the conditions are expressed in four rules: the dominant rules that form the dominant model – Model M – as displayed in Table 3-3.

Table 3-3: Dominant model of class 1 for MONK’s dataset (Model M)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1: If $X_1 = 1$ and $X_3 = 1$ then $Y = 1$</td>
<td>the robot has a round head shape and a round body shape</td>
</tr>
<tr>
<td>M2: If $X_2 = 1$ and $X_4 = 1$ then $Y = 1$</td>
<td>the robot has a square head shape and a square body shape</td>
</tr>
<tr>
<td>M3: If $X_1 = 0$, $X_2 = 0$, $X_3 = 0$ and $X_4 = 0$, then $Y = 1$</td>
<td>the robot has an octagonal head shape and an octagonal body shape</td>
</tr>
<tr>
<td>M4: If $X_8 = 1$ then $Y = 1$</td>
<td>the robot wears a red jacket</td>
</tr>
</tbody>
</table>

These four rules are the focus of all the existing studies, as the benchmark for comparisons on goodness of models and criteria.

Consider Model R, as shown in Table 3-4. There are seven rules, none of which appears in Table 3-3. Thus, by any existing criteria this is not a good model. However, it can be easily verified that this model will generate identical responses, as shown in Table 3-3. In other words, Model R is equivalent to Model M (the so-called true model). If any given rules selection technique selects Model R to represent the MONK’s dataset, to consider the technique incompetent would be a mistake. The seven rules from Model R do not appear in the dominant model, but they do contribute significantly to the other equivalent model. Treating them irrelevant would be a mistake as well.

Table 3-4: Equivalent model for MONK’s dataset (Model R)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1: If $X_8 = 1$ and $X_1 = 1$ then $Y = 1$</td>
<td>the robot (i) wears a red jacket and (ii) has a round head shape</td>
</tr>
<tr>
<td>R2: If $X_8 = 0$, $X_1 = 1$ and $X_3 = 1$ then $Y = 1$</td>
<td>the robot (i) has a round head shape, (ii) has a round body shape and (iii) does not wear a red jacket</td>
</tr>
<tr>
<td>R3: If $X_8 = 1$ and $X_4 = 1$ then $Y = 1$</td>
<td>the robot (i) wears a red jacket and (ii) the body shape is square</td>
</tr>
<tr>
<td>R4: If $X_8 = 0$, $X_2 = 1$, and $X_4 = 1$ then $Y = 1$</td>
<td>the robot (i) has a square head shape, (ii) has a square body shape and (iii) does not wear a red jacket</td>
</tr>
</tbody>
</table>
If \( X_1 = 0, X_3 = 0 \) and \( X_8 = 1 \) then \( Y = 1 \)

the robot (i) does not have a round head shape, (ii) does not have a round body shape and (iii) wears a red jacket

If \( X_1 = 0, X_2 = 0, X_3 = 0, X_4 = 0 \) and \( X_8 = 0 \) then \( Y = 1 \)

the robot (i) has an octagonal head shape, (ii) has an octagonal body shape and (iii) does not wear the red jacket

If \( X_1 = 0, X_4 = 0 \) and \( X_8 = 1 \) then \( Y = 1 \)

the robot (i) has a square head shape, (ii) has a square body shape and (iii) wears a red jacket

These two models are equivalent in that they generate identical responses. However, as shown in Tables 3-3 and 3-4, they explain the relationship between input variables and responses differently. Following the procedure discussed in Section 3.4, we next derive the dominant model for Model R.

### 3.5.2 Dominant Model for MONK’s Dataset

For Model R, the original seven rules cannot be deleted for the first consideration; therefore, we consider the irrelevant term. The term that can be omitted from Rule R1 is \( X_1 = 1 \), which leaves only term \( X_8 = 1 \) for the modified Rule R1. Then, Rule R3 can be deleted in the next step, as it becomes the child rule of the modified Rule R1. Next, the term that can be omitted is \( X_8 = 0 \) from Rule R2, which leaves only term \( X_1 = 1 \) and \( X_3 = 1 \) for the modified Rule R2. Then, Rule R5 can be deleted in the next step, as it becomes the child rule of the modified Rule R1. Next, the term that can be omitted is \( X_8 = 0 \) from Rule R4, which leaves only term \( X_2 = 1 \) and \( X_4 = 1 \) for the modified Rule R4. Then, Rule R7 can be deleted in the next step, as it becomes the child rule of the modified Rule R1. Next, the term that can be omitted is \( X_8 = 0 \) from Rule R6, which leaves only the term \( X_1 = 0, X_2 = 0, X_3 = 0, \) and \( X_4 = 0 \) for the modified Rule R6.

At this point, the dominant model, Model M, is obtained. Note that the modified Rule R1 is identical to Rule M4, the modified Rule R2 is identical to Rule M1, the modified Rule R4 is identical to Rule M2, and the modified Rule R6 is identical to Rule M3. The procedure for obtaining the dominant model for Model R is shown in Figure 3-5. From Theorem 1, this dominant model is unique.
Figure 3-5: Procedure for obtaining the dominant model for Model R

1. Model R
   - Delete term $X_1 = 1$ from Rule R1
   - Delete Rule R3
   - Delete term $X_8 = 0$ from Rule R2
   - Delete Rule R5
   - Delete term $X_8 = 0$ from Rule R4
   - Delete Rule R7
   - Delete term $X_8 = 0$ from Rule R6
   - Dominant model (Model M)
3.6 Conclusions

Association rules analysis typically considers a given model as the true model and ignores all other possibilities. The idea of models that generate the same dataset has never been adequately considered. The problem of such an equivalency occurs for any dataset. Moreover, this problem leads to misleading results in methodology development and comparison.

This study explores the issue of equivalent models in a comprehensive way. Not only do we focus on preventing misleading results, but we also provide the solution for unstable explanations from equivalent models. We propose using the benchmark model (the dominant model) as the representative for explaining the relationship between variables. The derivative of the benchmark model is provided. By using a well-known dataset, we emphasize that equivalence is sensitive to previous studies in association rules analysis. We hope that this discovery will provide a new aspect not only for association rules analysis but also for related data analysis methodologies.
Chapter 4

Analysis and Comparison for Rules selection Criteria

In this chapter, we study rules selection criteria in order to recommend the appropriate character of the rules selection criteria for the scenarios of interest, all of which involve interactions among the variables. The development and application of rules selection analysis are among the most intensively studied aspects in association rules analysis, such that a large number of rules selection criteria are available in the market. It is difficult to determine which criteria are the best as they have advantages and disadvantages depending on the circumstances. However, it is possible to determine the optimal criteria, on that is robust in a number of different scenarios, through systematic simulations.

In this chapter, we develop the methodological framework for rules selection comparisons. The framework is designed to generate detailed circumstances for different scenarios. We combine the selection of measurements in major studies and using a methodical simulation process compare their performances in capturing true models. The goal is to select the appropriate criteria that are robust for selecting true models for a variety of scenarios. We also consider the issue of equivalent sets of rules for the categorical variables. The analysis of the performance comparisons and the relationships among these criteria is also thoroughly investigated.

The two rules selection criteria introduced initially with the association rules analysis are support and confidence. Rules that meet the standards of minimum support and minimum confidence are considered to be important in this context. After the introduction of association rules analysis, many research studies have focused on improving this technique. The one obvious way is to apply alternative rules selection criteria with association rules analysis. From a thorough search of the data mining papers and journals in the recent literature, almost 50 rules selection criteria have been applied or developed to date (Tan et al. 2004; Geng et al. 2006; Blanchard et al. 2009). Among these criteria, comparisons have been made to find relationships between and establish the properties of all the criteria. In previous work on criteria comparison,
Ohsaki et al. (2004) and Tan et al. (2004) compare rankings by experts and by rules selection criteria for specific datasets. Lenca et al. (2004) provide a criteria selection method by using a multicriteria decision strategy. The properties for each criterion is provided and the selected criteria are based on the properties desired by users. The practitioner needs to weigh the significance of each property for rules selection criteria. Vaillant et al. (2004) cluster rules selection criteria into groups based on their properties and their rankings based on a specified rule set. Geng and Hamilton (2006) survey the rules selection criteria and analyze them according to several perspectives including summarizing their properties and objectives. However, in this study, we compare and analyze the rules selection criteria based on their performances in our experiment, wherein we use general datasets from the scenarios of interest and do so without depending on the judgments of experts.

Accordingly, we develop a methodological framework for rules selection comparisons. The framework is designed to systematically generate detailed circumstances for the scenarios of interest, i.e., scenarios in which the true models consist of the interactions among the variables. The goal is to use a methodical simulation process to compare the performances of the rules selection criteria in capturing true models. The objectives are to recommend the criteria that are robust in selecting true models for a variety of scenarios and to analyze and compare the properties for criteria according to their performances.

This chapter is organized as follows. Section 4.1 is a review rules selection criteria. Section 4.2 presents the simulation process and comparison procedure. Section 4.3 demonstrates the simulation results and analysis. Section 4.4 offers a discussion and concluding remarks.

### 4.1 Rules Selection Criteria

In association rules analysis, there are a large number of rules selection criteria in the literature. From a thorough search of the data mining papers and journals in the recent literature, 46 criteria are summarized, as shown in Table 4-1, which also presents detailed notations. These 46 criteria are also illustrated in one of the three summarized papers: Tan et al. (2004), Geng et al. (2006), and Blanchard et al. (2009).

Among the 46 criteria, 42 measures do not need setting coefficients and can be re-written by using three variables, confidence \(c\), support \(s\), and probability of class or result \(P(B)\). Note
that Yule’s Q and Goodman and Kruskal are the same for a 2 by 2 table (Agresi 2002). Therefore, we omit Goodman and Kruskal and use Yule’s Q to represent this measurement. Moreover, Loevinger is equal to 1-conviction. Therefore, we omit Loevinger, and the total number of criteria in consideration becomes 40. We analyze the performance of these 40 criteria with the scenarios of interest.
Table 4-1: Rules Selection Criteria summarized from Tan et. al 2004, Geng et al. 2006, and Blanchard et. al 2009. (order alphabetically)

For the rule if A then B, $P(A)$ is the probability of condition (A), $P(B)$ is the probability of result (B), $P(AB) = P(A \cap B)$, $P(A'B) = P(A' \cap B)$, $P(AB') = P(A \cap B')$, support $(s) = P(AB)$ and confidence $(c) = P(AB)/P(A)$.

<table>
<thead>
<tr>
<th>No.</th>
<th>Measure</th>
<th>Formula</th>
<th>Rewritten Formula</th>
<th>Tan</th>
<th>Geng</th>
<th>Blanchard</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Accuracy</td>
<td>$P(AB) + P(A'B')$</td>
<td>$1 - \frac{s}{c} - P(B) + 2s$</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>Added value</td>
<td>$P(B</td>
<td>A) - P(B)$</td>
<td>$c - P(B)$</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>Certainty factor</td>
<td>$\frac{P(B</td>
<td>A) - P(B)}{1 - P(B)}$</td>
<td>$\frac{c - P(B)}{1 - P(B)}$</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>Confidence</td>
<td>$\frac{P(AB)}{P(A)}$</td>
<td>$c$</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>Conviction</td>
<td>$\frac{P(A)P(B')}{P(AB')}$</td>
<td>$\frac{1 - P(B)}{1 - c}$</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>6</td>
<td>Collective strength</td>
<td>$\left( \frac{P(AB) + P(A'B')}{P(A)P(B) + P(A')P(B')} \right) \times \left( \frac{1 - P(AB) - P(A'B')}{} \right)$</td>
<td>$\left( \frac{1 + 2s - \frac{s}{c} - P(B)}{1 + \frac{2sp(B)}{c} - \frac{s}{c} - P(B)} \right) \times \left( \frac{s + P(B) - 2sp(B)}{c} \right)$</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>7</td>
<td>Cosine</td>
<td>$\frac{P(AB)}{\sqrt{P(A)P(B)}}$</td>
<td>$\frac{sc}{P(B)}$</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>Dice index</td>
<td>$\frac{P(AB)}{P(AB) + \frac{1}{2}(P(A'B) + P(AB'))}$</td>
<td>$\frac{cs}{cs + \frac{1}{2}cP(B) + \frac{1}{2}s}$</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>9</td>
<td>Directed contribution to Chi-squared</td>
<td>$\sqrt{\frac{(P(AB) - P(A)P(B))}{P(A)P(B)}}$</td>
<td>$\sqrt{\frac{ncs}{c}} \left( 1 - \frac{P(B)}{c} \right)$</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>10</td>
<td>Example and counterexample rate</td>
<td>$1 - \frac{P(AB')}{P(AB)}$</td>
<td>$2 - \frac{1}{c}$</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>11</td>
<td>Ganascia index</td>
<td>$\frac{P(AB) - P(AB')}{P(A)}$</td>
<td>$2c - 1$</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>No.</td>
<td>Description</td>
<td>Formula</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>---------------------------</td>
<td>----------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Gini index</td>
<td>[ P(A)(P(B</td>
<td>A)^2 + P(B'</td>
<td>A)^2) + P(A')(P(B</td>
<td>A')^2 + P(B'</td>
<td>A')^2) - P(B)^2 - P(B')^2 ]</td>
</tr>
<tr>
<td>13</td>
<td>Goodman and Kruskal</td>
<td>Same as Yule’s Q for 2*2 table (Agresti, 2002)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Implication index</td>
<td>[-\sqrt{N} \left( \frac{P(AB) - P(A)P(B')}{\sqrt{P(A)P(B')}} \right) ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Implication intensity</td>
<td>[ P(\text{Poisson}(NP(A)P(B')) &gt; NP(AB')) ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Inclusion index</td>
<td>[ \sqrt{\sum_{i=1}^{r} \sum_{j=1}^{s} \frac{i^2}{a_{ij}} \cdot \frac{j^2}{a_{ij}} = 0} ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Information gain</td>
<td>[ \log \left( \frac{P(AB)}{P(A)P(B)} \right) ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Interestingness weighting dependency</td>
<td>[ \left{ \frac{P(AB)}{P(A)P(B)} \right}^k \left( 1 - \left( \frac{P(AB)}{P(A)P(B)} \right)^m \right) ] , where k and m are coefficients of dependency and generality, respectively.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Jaccard</td>
<td>[ \frac{P(AB)}{(P(A) + P(B) - P(AB))} ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>J-measure</td>
<td>[ P(AB) \log \left( \frac{P(B</td>
<td>A)}{P(B)} \right) + P(AB') \log \left( \frac{P(B'</td>
<td>A)}{P(B')} \right) ]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Kappa</td>
<td>[ \frac{P(AB) + P(AB') - P(A)P(B) - P(A')P(B')}{1 - P(A)P(B) - P(A')P(B')} ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Klosgen</td>
<td>[ \sqrt{P(AB)(P(B</td>
<td>A) - P(B))} ]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Kuleczynski index</td>
<td>[ \frac{1}{2} \left( \frac{P(AB)}{P(A)} + \frac{P(AB)}{P(B)} \right) ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Laplace</td>
<td>[ \frac{N(AB) + 1}{N(A) + 2} ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Least contradiction</td>
<td>[ \frac{P(AB) - P(AB')}{P(B)} ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Leverage</td>
<td>( \left( P(B</td>
<td>A) - P(A)P(B) \right) )</td>
<td>( c - s ) ( P(B) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Lift</td>
<td>( \frac{P(AB)}{P(A)P(B)} )</td>
<td>( \frac{c}{P(B)} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Likelihood linkage index</td>
<td>( P(\text{Poisson}(NP(A)P(B)) &lt; NP(AB)) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Loewinger</td>
<td>( 1 - \frac{P(A)P(B')}{P(AB')} )</td>
<td>( 1 - \text{conviction} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Normalized mutual information</td>
<td>( \sum \sum \frac{P(A,B)}{P(A)P(B)} \log_2 \frac{P(A,B)}{P(A)P(B)} - \frac{1}{n} \sum \log_2 P(A) )</td>
<td>Two-way support</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Odd multiplier</td>
<td>( \frac{P(AB)P(B')}{P(B)P(AB')} )</td>
<td>( \frac{P(B)}{1 - P(B)} \times \frac{c}{1 - c} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>Odds ratio</td>
<td>( \frac{P(AB)P(A'B')}{P(A'B)P(AB')} )</td>
<td>( \frac{c(1 - P(B)) - s(1 - c)}{(1 - c)(P(B) - s)} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>One-way support</td>
<td>( P(B</td>
<td>A) \log_2 \left{ \frac{P(AB)}{P(A)P(B)} \right} )</td>
<td>( c \log_2 \left{ \frac{c}{P(B)} \right} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Piatetsky-Shapiro</td>
<td>( P(AB) - P(A)P(B) )</td>
<td>( s \left( 1 - \frac{P(B)}{c} \right) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>Recall</td>
<td>( P(A</td>
<td>B) )</td>
<td>( \frac{s}{P(B)} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>Relative risk</td>
<td>( \frac{P(B</td>
<td>A)}{P(B</td>
<td>A')} )</td>
<td>( \frac{(c - s)}{(P(B) - s)} )</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>Rogers-Tanimoto index</td>
<td>( \frac{1 - P(A'B') - P(AB')}{1 + P(A'B) + P(AB')} )</td>
<td>( \frac{1 - P(B) - s + 2s}{1 + P(B) + s - 2s} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Sebag-Schoenauer</td>
<td>( \frac{P(AB)}{P(AB')} )</td>
<td>( \frac{c}{1 - c} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>Specificity</td>
<td>( P(B'</td>
<td>A') )</td>
<td>( \frac{c - s + cs - cP(B)}{(c - s)} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>Support</td>
<td>( P(AB) )</td>
<td>( s )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Tau-b</td>
<td>( \sqrt{\frac{P(AB) - P(A)P(B)}{P(A)P(B)'P(A')P(B')}} )</td>
<td>( \sqrt{\frac{s(c - P(B))^2}{(c - s)P(B)(1 - P(B))}} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>Two-way Support</td>
<td>( P(AB) \log_2 \left{ \frac{P(AB)}{P(A)P(B)} \right} )</td>
<td>( s \log_2 \left{ \frac{c}{P(B)} \right} )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

49
<table>
<thead>
<tr>
<th></th>
<th>Two-way support variation</th>
<th></th>
<th>( J - \text{measure} + )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( P(AB) \log_2 \left( \frac{P(AB)}{P(A)P(B)} \right) + )</td>
<td></td>
<td>( (P(B) - s) \log_2 \left( \frac{c(P(B) - s)}{(c - s)P(B)} \right) + )</td>
</tr>
<tr>
<td></td>
<td>( P(AB') \log_2 \left( \frac{P(AB')}{P(A)P(B')} \right) + )</td>
<td></td>
<td>( (1 + s - \frac{s}{c} - P(B)) \log_2 \left( \frac{1 + s - \frac{s}{c} - P(B)}{1 - \frac{s}{c}(1 - P(B))} \right) )</td>
</tr>
<tr>
<td></td>
<td>( P(A'B) \log_2 \left( \frac{P(A'B)}{P(A')P(B)} \right) + )</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( P(A'B') \log_2 \left( \frac{P(A'B')}{P(A')P(B')} \right) )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Column 4 states the original formula. Column 5 states the re-written formula as form of support (s), confidence (c) and the probability of result (P(B)). The columns 6, 7, and 8 show a check mark if the corresponding criterion appears in Tan et al. (2004), Geng et al. (2006), and Blanchard et al. (2009), respectively.
4.2 Simulation Process and Comparison Procedure

We use simulations to compare the performance of the rules selection criteria. By using simulations, we can observe their ability to find the true or underlined model for each criterion, thus to and select criteria that are robust with the scenarios of interest, i.e., scenarios in the interactions are involved in the true models. The simulation is designed to be general enough to allow fair conclusions to be drawn and guidelines for criteria selection to be established. This section presents detailed information about the design matrix (Section 4.2.1), the simulation procedures (Section 4.2.2), and the comparison procedure (Section 4.2.3). It also offers an example of the simulation process (Section 4.2.4).

4.2.1 Details of the Design Matrix

There are two design matrices for this study: (1) a full factorial design for 6 binary variables, referred to as 2(6) with a total of 64 observations, and (2) a full factorial design for 10 binary variables, referred to as 2(10) with a total of 1,024 observations. We use these two design matrices mainly to study changes in the performances of criteria when the number of variables changes in the experiments. The example of the first design matrix is given in Table 4-4. We use the full factorial design for our study, as it covers all the possible combinations among the independent variables. It is fair from the sampling viewpoint. There is no bias in this design that will play a role in criteria comparison. Moreover, the design can easily be modified by adding variables or eliminating some variables. Note that the full factorial design is also used in MONK’s dataset.

4.2.2 Simulation Process

The simulation is separated into two parts. The first part generates the datasets. The second part tests the performances of the rules selection criteria with the generated datasets. For the first part, the data-generating procedure is shown in Figure 4-1. Note that the procedure shows how to generate datasets for one specific scenario. However, a similar procedure will be followed for the other scenarios, which will be shown in the follow sections.
The First Part: Data Generation Procedure

This is the first part of the simulation procedure. It is designed to generate responses from the design matrix.

![Diagram of data generation procedure for two-rule model]

Figure 4-1: Data generation procedure for two-rule model

Details of the data generation procedure are given below.

**Step 1:** From the design matrix, observe the distinctive 2-rule models for the scenarios of interest. The two-rule model consists of two rules. For convenience, we use class 1 as the results for all the models. The scenarios of interest are as shown below.

**Scenario 1:** One rule contains only one variable in the condition and the other contains two variables in the condition, referred to as **main effect + two-way interaction** models

**Scenario 2:** One rule contains only one variable in the condition and the other contains three variables in the condition, referred to as **main effect + three-way interaction** models

**Scenario 3:** One rule contains two variables in the condition and the other also contains two variables in the condition, referred to as **two-way interaction + two-way interaction** models

**Scenario 4:** One rule contains two variables in the condition and the other contains three variables in the condition, referred to as **two-way interaction + three-way interaction** models

**Scenario 5:** One rule contains three variables in the condition and the other contains three variables in the condition, referred to as **three-way interaction + three-way interaction** models
**Step 2:** For each model, generate response Y accordingly.

**Step 3:** Combine the response Y with the input matrix to generate the dataset. If there are k models, there will be k datasets.

For the five scenarios in Table 4-1, there are 14 initial models as shown in Table 4-2.

**Table 4-2: The 14 initial models for the two-rule model**

<table>
<thead>
<tr>
<th>Initial model</th>
<th>Condition</th>
<th>Result</th>
<th>Can the model be reduced?</th>
<th>Condition of the reduced form</th>
<th>Result of the reduced form</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>X1 = 0</td>
<td>Y = 1</td>
<td>Yes</td>
<td>X1 = 0</td>
<td>Y = 1</td>
</tr>
<tr>
<td></td>
<td>X1 = 1, X2 = 0</td>
<td>Y = 1</td>
<td></td>
<td>X2 = 0</td>
<td>Y = 1</td>
</tr>
<tr>
<td>ii</td>
<td>X1 = 0</td>
<td>Y = 1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X2 = 0, X3 = 0</td>
<td>Y = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iii</td>
<td>X1 = 0</td>
<td>Y = 1</td>
<td>Yes</td>
<td>X1 = 0</td>
<td>Y = 1</td>
</tr>
<tr>
<td></td>
<td>X1 = 1, X2 = 0, X3 = 0</td>
<td>Y = 1</td>
<td></td>
<td>X2 = 0, X3 = 0</td>
<td>Y = 1</td>
</tr>
<tr>
<td>iv</td>
<td>X1 = 0</td>
<td>Y = 1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X2 = 0, X3 = 0, X4 = 0</td>
<td>Y = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v</td>
<td>X1 = 0, X2 = 0</td>
<td>Y = 1</td>
<td>Yes</td>
<td>X1 = 0</td>
<td>Y = 1</td>
</tr>
<tr>
<td></td>
<td>X1 = 0, X2 = 1</td>
<td>Y = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vi</td>
<td>X1 = 0, X2 = 0</td>
<td>Y = 1</td>
<td>No</td>
<td>However, the result will be the same as model ii as explained below</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X1 = 0, X3 = 0</td>
<td>Y = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vii</td>
<td>X1 = 0, X2 = 0</td>
<td>Y = 1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X3 = 0, X4 = 0</td>
<td>Y = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>viii</td>
<td>X1 = 0, X2 = 0</td>
<td>Y = 1</td>
<td>Yes</td>
<td>X1 = 0, X2 = 0</td>
<td>Y = 1</td>
</tr>
<tr>
<td></td>
<td>X1 = 0, X2 = 1, X3 = 0</td>
<td>Y = 1</td>
<td></td>
<td>X1 = 0, X3 = 0</td>
<td>Y = 1</td>
</tr>
<tr>
<td>ix</td>
<td>X1 = 0, X2 = 0</td>
<td>Y = 1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X1 = 0, X3 = 0, X4 = 0</td>
<td>Y = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>X1 = 0, X2 = 0</td>
<td>Y = 1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X3 = 0, X4 = 0, X5 = 0</td>
<td>Y = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xi</td>
<td>X1 = 0, X2 = 0, X3 = 0</td>
<td>Y = 1</td>
<td>Yes</td>
<td>X1 = 0, X2 = 0</td>
<td>Y = 1</td>
</tr>
<tr>
<td></td>
<td>X1 = 0, X2 = 0, X3 = 1</td>
<td>Y = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xii</td>
<td>X1 = 0, X2 = 0, X3 = 0</td>
<td>Y = 1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X1 = 0, X2 = 0, X4 = 0</td>
<td>Y = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xiii</td>
<td>X1 = 0, X2 = 0, X3 = 0</td>
<td>Y = 1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X1 = 0, X4 = 0, X5 = 0</td>
<td>Y = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xiv</td>
<td>X1 = 0, X2 = 0, X3 = 0</td>
<td>Y = 1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X4 = 0, X5 = 0, X6 = 0</td>
<td>Y = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An example of how to read the table is illustrated via model i. For model i, the two rules are (1) if X1 = 0 then Y = 1 and (2) if X1 = 0 and X2 = 0 then Y = 1. With these two rules, the response is generated accordingly. However, this model can omit some variables
but still generate an identical response. The simpler form of this mode consists of the following two rules: (1) if \( X_1 = 0 \) then \( Y = 1 \) and (2) if \( X_2 = 0 \) then \( Y = 1 \). The reference can be found in Changpetch and Lin, 2012. According to the simpler form, this model no longer belongs to the scenarios of interest, which need at least one rule wherein variables interact. Models i, v, and xi are the models for which the reduced forms do not belong to the scenarios of interest.

In addition, models ii, vi, and viii, all yield the same result. Model ii and model vi are the same because the complement model for model vi comprises two rules: (1*) if \( X_1 = 1 \) then \( Y = 0 \) and (2*) if \( X_2 = 1 \) and \( X_3 = 1 \) then \( Y = 0 \); therefore, switching the number between zero and one in the conditions and results yields model ii. On the other hand, the complement model of model ii is (1#) if \( X_1 = 1 \) and \( X_2 = 1 \) then \( Y = 0 \) and (2#) if \( X_1 = 1 \) and \( X_3 = 1 \) then \( Y = 0 \); therefore, switching the number between zero and one in the conditions and results yields model vi. Therefore, these two models yield the same result, as if zero and one are switched but the result is the same. As model viii can be reduced to model vi, model viii will yield the same result as model ii. Therefore, we can omit model vi and model viii from further analysis.

In consequence, we consider only the models that cannot be reduced to a simpler form. We consider the models shown in Table 4-3.
Table 4-3: The final models of our interest.

<table>
<thead>
<tr>
<th>Model</th>
<th>Given Rules</th>
<th>Complement Model</th>
<th>Rules in complement Model</th>
</tr>
</thead>
</table>
| Model 1 | 1a: \(X_1 = 0 \rightarrow Y = 1\)  
1b: \(X_2 = 0, X_3 = 0 \rightarrow Y = 1\) | Model 1' | 1'a: \(X_1 = 1, X_2 = 1 \rightarrow Y = 0\)  
1'b: \(X_1 = 1, X_3 = 1 \rightarrow Y = 0\) |
| Model 2 | 2a: \(X_1 = 0 \rightarrow Y = 1\)  
2b: \(X_2 = 0, X_3 = 0, X_4 = 0 \rightarrow Y = 1\) | Model 2' | 2'a: \(X_1 = 1, X_2 = 1 \rightarrow Y = 0\)  
2'b: \(X_1 = 1, X_3 = 1 \rightarrow Y = 0\)  
2'c: \(X_1 = 1, X_4 = 1 \rightarrow Y = 0\) |
| Model 3 | 3a: \(X_1 = 0, X_2 = 0 \rightarrow Y = 1\)  
3b: \(X_3 = 0, X_4 = 0 \rightarrow Y = 1\) | Model 3' | 3'a: \(X_1 = 1, X_3 = 1 \rightarrow Y = 0\)  
3'b: \(X_1 = 1, X_4 = 1 \rightarrow Y = 0\)  
3'c: \(X_2 = 1, X_3 = 1 \rightarrow Y = 0\)  
3'd: \(X_2 = 1, X_4 = 1 \rightarrow Y = 0\) |
| Model 4 | 4a: \(X_1 = 0, X_2 = 0 \rightarrow Y = 1\)  
4b: \(X_1 = 0, X_3 = 0, X_4 = 0 \rightarrow Y = 1\) | Model 4' | 4'a: \(X_1 = 1 \rightarrow Y = 0\)  
4'b: \(X_2 = 1, X_3 = 1 \rightarrow Y = 0\)  
4'c: \(X_2 = 1, X_4 = 1 \rightarrow Y = 0\) |
| Model 5 | 5a: \(X_1 = 0, X_2 = 0 \rightarrow Y = 1\)  
5b: \(X_3 = 0, X_4 = 0, X_5 = 0 \rightarrow Y = 1\) | Model 5' | 5'a: \(X_1 = 1, X_3 = 1 \rightarrow Y = 0\)  
5'b: \(X_1 = 1, X_4 = 1 \rightarrow Y = 0\)  
5'c: \(X_1 = 1, X_5 = 1 \rightarrow Y = 0\)  
5'd: \(X_2 = 1, X_3 = 1 \rightarrow Y = 0\)  
5'e: \(X_2 = 1, X_4 = 1 \rightarrow Y = 0\)  
5'f: \(X_2 = 1, X_5 = 1 \rightarrow Y = 0\) |
| Model 6 | 6a: \(X_1 = 0, X_2 = 0, X_3 = 0 \rightarrow Y = 1\)  
6b: \(X_1 = 0, X_2 = 0, X_4 = 0 \rightarrow Y = 1\) | Model 6' | 6'a: \(X_1 = 1 \rightarrow Y = 0\)  
6'b: \(X_2 = 1 \rightarrow Y = 0\)  
6'c: \(X_3 = 1, X_4 = 1 \rightarrow Y = 0\) |
| Model 7 | 7a: \(X_1 = 0, X_2 = 0, X_3 = 0 \rightarrow Y = 1\)  
7b: \(X_1 = 0, X_4 = 0, X_5 = 0 \rightarrow Y = 1\) | Model 7' | 7'a: \(X_1 = 1 \rightarrow Y = 0\)  
7'b: \(X_2 = 1, X_4 = 1 \rightarrow Y = 0\)  
7'c: \(X_2 = 1, X_5 = 1 \rightarrow Y = 0\)  
7'd: \(X_3 = 1, X_4 = 1 \rightarrow Y = 0\)  
7'e: \(X_3 = 1, X_5 = 1 \rightarrow Y = 0\) |
| Model 8 | 8a: \(X_1 = 0, X_2 = 0, X_3 = 0 \rightarrow Y = 1\)  
8b: \(X_4 = 0, X_5 = 0, X_6 = 0 \rightarrow Y = 1\) | Model 8' | 8'a: \(X_1 = 1, X_4 = 1 \rightarrow Y = 0\)  
8'b: \(X_1 = 1, X_5 = 1 \rightarrow Y = 0\)  
8'c: \(X_1 = 1, X_6 = 1 \rightarrow Y = 0\)  
8'd: \(X_2 = 1, X_4 = 1 \rightarrow Y = 0\)  
8'e: \(X_2 = 1, X_4 = 1 \rightarrow Y = 0\)  
8'f: \(X_2 = 1, X_5 = 1 \rightarrow Y = 0\)  
8'g: \(X_3 = 1, X_4 = 1 \rightarrow Y = 0\)  
8'h: \(X_3 = 1, X_5 = 1 \rightarrow Y = 0\)  
8'i: \(X_3 = 1, X_4 = 1 \rightarrow Y = 0\) |

The eight models under consideration each generate one dataset for a total of eight datatsets, all of which will be used in the next procedure.
The Second Part: Performance Testing Procedure

The second part of the simulation process is to test the performance of each rules selection criterion. The main goal is to observe each method’s ability to search for the true models for the generated datasets. The procedure is shown in Figure 4-2.

The testing procedure is dividing into two schemes. The difference between the two schemes arises at step 2. Details of the performance testing procedure for the first scheme are described next.
First Scheme

Step 1: For each dataset, the criterion will generate all the rules from the main effect to four-way interactions. The rule with the main effect contains only one variable in the condition, whereas the rule with the four-way interactions has four variables in the condition.

Step 2a: Among all those rules, rank them by the values of the criterion.

Step 3: Search for the true response from rule 1 to the higher (worse) ranked rules. The rules that are unnecessary to or contradict the dataset will be thrown away.

Step 4: Record the number of rules needed to obtain the true response. For example, we search for the true response from rule 1 to rule 10. The number 10 will be recorded as the number of rules needed. If there are ties, the highest number of rules with all the ties combined will be recorded.

Step 5: From all the datasets in each scenario, summarize the number of rules needed to obtain the true responses.

Note that we repeat this process for all the scenarios and all the rules selection criteria.

Second Scheme

Step 1: For each dataset, the criterion will generate all the rules from the main effect to the four-way interactions.

Step 2b: Among all those rules, rank them by the following order:

i) Value of the criterion

ii) Value of support

iii) Rule size (the number of variables in the condition)—the smaller the rule size, the better

Step 3: Search for the true response from rule 1 to the higher (worse) ranked rules. The rules that are unnecessary to or contradict the dataset will be thrown away.

Step 4: Record the number of rules needed to obtain the true response.

Step 5: From all the datasets in each scenario, summarize the number of rules needed to obtain the true responses.

The summary of the simulation process is shown in Figure 4-3.
Each criterion applies to both design matrices. From Figure 4-3a, we summarize the simulation process for the design matrix with 6 variables 2(6). For this design matrix, the data for eight models in Table 4-3 are generated. Each criterion will be tested with all eight models and the numbers of rules needed to find the true models are recorded. For this design matrix, we conduct the performance testing according to two schemes: the first scheme does not use support or rule size to break ties in step 2, whereas the second scheme uses support and rule size to break ties in step 2.

In Figure 4-3b, we summarize the simulation process for the design matrix with 10 variables 2(10). Similar to Figure 4-1, the data for the eight models in Table 4-3 are generated in the data-generating procedure. Each criterion will be tested with all eight models, and the numbers of rules needed to find the true models recorded. For this design matrix, we perform only the second scheme for the criteria comparison. We do not apply the second design matrix with the second scheme, as this is clarified in the next section.
4.2.3 Comparison Procedure

We offer two main comparisons in our study. First, we compare the performances of all the rules selection criteria when support and rule size are involved and when support and rule size are not involved. We apply all the criteria with the first design matrix 2(6) for both schemes for this purpose. As the first scheme does not use support and rule size to break ties and the second scheme does use support and rule size to break ties, we can observe which criteria depend on support and rule size to improve their performance. In other words, we can determine which criteria depend more on support and rule size in breaking ties.

Second, we compare the performance of all the criteria when the number of variables increases. We apply all the criteria with the second design matrix 2(10) with the second scheme and we do the same with the performance of the 2(6) in the second scheme. We then compare the behaviors of all the criteria when the number of variables increases.

4.2.4 Illustrated Example for Simulation Process

In this section, we give an example to show how we obtain the number from the simulation process. We use the example of the first design matrix with 6 variables 2(6) with Model 1 from Table 4-3.

After the data have been generated, the response from Model 1 combined with the design matrix become dataset 1, as shown in Table 4-4.

Table 4-4: Data generation for design matrix 2(6) using Model 1

<table>
<thead>
<tr>
<th>Design Matrix</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1 X2 X3 X4 X5 x6</td>
<td>Y</td>
</tr>
<tr>
<td>1 1 1 1 1 1</td>
<td>0</td>
</tr>
<tr>
<td>0 1 1 1 1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 0 1 1 1 1</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1 1 1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1 0 1 1 1</td>
<td>0</td>
</tr>
<tr>
<td>0 1 0 1 1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 0 0 1 1 1</td>
<td>1</td>
</tr>
<tr>
<td>0 0 0 1 1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1 1 0 1 1</td>
<td>0</td>
</tr>
<tr>
<td>0 1 1 0 1 1</td>
<td>0</td>
</tr>
<tr>
<td>1 0 1 0 1 1</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1 0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1 0 0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>0 1 0 0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1 1 1 0 1</td>
<td>1</td>
</tr>
<tr>
<td>0 1 1 1 0 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1 0 1 0 1</td>
<td>1</td>
</tr>
<tr>
<td>0 1 0 1 0 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1 0 0 0 1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Dataset 1 from Table 4-4 is the input for the performance testing procedure, the generated rules and their ranks according to the confidence criterion are shown in Table 4-5. Note that we show only the top 30 rules.

<table>
<thead>
<tr>
<th>Rule no.</th>
<th>Rank (scheme 1)</th>
<th>Rank (scheme 2)</th>
<th>Condition</th>
<th>Response</th>
<th>Confidence</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>X1(0)</td>
<td>Y(1)</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>X1(1)X2(1)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>X1(1)X3(1)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>X1(0)X2(0)</td>
<td>Y(1)</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>X1(0)X2(1)</td>
<td>Y(1)</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>X1(0)X3(0)</td>
<td>Y(1)</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>X1(0)X3(1)</td>
<td>Y(1)</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2</td>
<td>X1(0)X4(0)</td>
<td>Y(1)</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2</td>
<td>X1(0)X4(1)</td>
<td>Y(1)</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2</td>
<td>X1(0)X5(0)</td>
<td>Y(1)</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>2</td>
<td>X1(0)X5(1)</td>
<td>Y(1)</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>2</td>
<td>X1(0)X6(0)</td>
<td>Y(1)</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>2</td>
<td>X1(0)X6(1)</td>
<td>Y(1)</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>2</td>
<td>X2(0)X3(0)</td>
<td>Y(1)</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>15</td>
<td>X1(1)X2(1)X3(0)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>15</td>
<td>X1(1)X2(0)X3(1)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>15</td>
<td>X1(1)X2(1)X3(1)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>15</td>
<td>X1(1)X2(1)X4(0)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>15</td>
<td>X1(1)X2(1)X4(1)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>15</td>
<td>X1(1)X2(1)X5(0)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>15</td>
<td>X1(1)X2(1)X5(1)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>15</td>
<td>X1(1)X2(1)X6(0)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>15</td>
<td>X1(1)X2(1)X6(1)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>15</td>
<td>X1(1)X3(1)X4(0)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>15</td>
<td>X1(1)X3(1)X4(1)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>15</td>
<td>X1(1)X3(1)X5(0)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td>27</td>
<td>1</td>
<td>15</td>
<td>X1(1)X3(1)X5(1)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>15</td>
<td>X1(1)X3(1)X6(0)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>15</td>
<td>X1(1)X3(1)X6(1)</td>
<td>Y(0)</td>
<td>1</td>
<td>0.125</td>
</tr>
</tbody>
</table>

Table 4-5: Generated rules and their rankings for the confidence criterion
Note: The first column represents the number of the rule. The second column represents the rank of its corresponding rule based on scheme 1 (not using support and rule size to break ties). The third column represents the rank of its corresponding rule based on scheme 2 (using support and rule size to break ties). The fourth column represents the condition of the rule, e.g., X1(0)X2(0) means X1 = 0 and X2 = 0. The fifth column represents the result for the rule, Y(0) means Y = 0 and Y(1) means Y = 1. The sixth column represents the value of the criterion. The last column represents the value of support.

From Table 4-5, using scheme 2 for the performance testing procedure, there is 1 rule for rank 1, which is if X1 = 0 then Y = 1 with confidence = 1, support = 0.5, and rule size = 1. Thirteen rules are ranked 2, as they all have confidence = 1, support = 0.25, and rule size = 2. To find the true model, we search from rule no. 1 to the higher-ranked rules. With this example, the rules that can generate the true model are a combination of rule no. 1 and rule no. 14 and a combination of rule no. 2 and rule no. 3. For scheme 2, for the first combination, the number of rules needed is 14, as the second rule is attached to the other 12 rules. For the second combination, the number of rules needed is also 14, as the two rules are both ranked 2 with the other 11 rules attached. Therefore, the number of rules needed for this scheme is 14. For scheme 1, the number of rules needed is from the number of rules with confidence = 1 which is 216 rules. As ties cannot be broken, all the attached rules with the same rank must be added together in order to count the number of rules needed.

Let us see the other example for the same dataset (dataset1) using the J-measure to find the true model. The generated rules and their ranks according to the J-measure criterion are shown in Table 4-6.

**Table 4-6: Generated rules and their rankings for the J-measure criterion**

<table>
<thead>
<tr>
<th>Rule no.</th>
<th>Rank (scheme 1)</th>
<th>Rank (scheme 2)</th>
<th>Condition</th>
<th>Response</th>
<th>Implication index</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>X1(1)X2(1)</td>
<td>Y(0)</td>
<td>0.106492</td>
<td>0.25</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>X1(1)X3(1)</td>
<td>Y(0)</td>
<td>0.106492</td>
<td>0.25</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>X1(0)</td>
<td>Y(1)</td>
<td>0.10206</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>X1(1)</td>
<td>Y(0)</td>
<td>0.063144</td>
<td>0.375</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>X1(1)</td>
<td>Y(1)</td>
<td>0.063144</td>
<td>0.125</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>X1(1)X2(1)X3(0)</td>
<td>Y(0)</td>
<td>0.053246</td>
<td>0.125</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>6</td>
<td>X1(1)X2(0)X3(1)</td>
<td>Y(0)</td>
<td>0.053246</td>
<td>0.125</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>6</td>
<td>X1(1)X2(1)X3(1)</td>
<td>Y(0)</td>
<td>0.053246</td>
<td>0.125</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>6</td>
<td>X1(1)X2(1)X4(0)</td>
<td>Y(0)</td>
<td>0.053246</td>
<td>0.125</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>6</td>
<td>X1(1)X2(1)X4(1)</td>
<td>Y(0)</td>
<td>0.053246</td>
<td>0.125</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>6</td>
<td>X1(1)X2(1)X5(0)</td>
<td>Y(0)</td>
<td>0.053246</td>
<td>0.125</td>
</tr>
</tbody>
</table>
From Table 4-6, the rankings for scheme 1 and scheme 2 are the same, which means that this criterion does not need support and rule size to break ties or to improve its performance. To find the true model, we search from rule no. 1 to the higher-ranked rules. With this example, the first two rules combined can generate the true model; therefore, the number of rules needed for the J-measure with this example is 2.

### 4.3 Performance Comparison and Analysis

There are two comparisons for this section. The first comparison focuses on finding out which criteria depend on support and rule size in their performance, specifically to help in break ties. The second comparison focuses on analyzing the behavior of each criterion when the number of variables increases. The performance comparisons are illustrated with bar charts in Appendix A and Appendix B, and the complete simulation results are shown in Appendix C.
4.3.1 The Comparison for Dependency on Support and Rule Size in Breaking Ties

The first comparison is performed to find out which of the criteria are best at addressing ties. In other words, we determine which criteria do not need support and rule size to break ties. Therefore, we compare the performances of each criterion with and without using support and rule size. For this purpose, we use the design matrix with 6 binary variables. The performance comparisons are illustrated with bar charts in Appendix A and summarized in Table 4-7.

Table 4-7: Are there any changes in the number of rules needed with and without support and rule size?

<table>
<thead>
<tr>
<th>Criteria\Model</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>accur</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>added</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>certa</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>conf</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>convi</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>coole</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cosin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dice</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>direc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>examp</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>ganas</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>gini</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>impli</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>infor</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>jacc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jmeas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kappa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>klosq</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kulcz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lapla</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>least</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lever</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lift</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>norma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>oddmu</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>oddra</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>onesu</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>recal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>relat</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>roger</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sebag</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>speci</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>taub</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
From the results comparison, 15 criteria need support and confidence to improve their performance. In other words, 15 criteria need support and rule size to help break ties. These 15 criteria can be divided further into three groups based on the same number of rules needed in all the models (see Appendix A). The three groups are shown in Table 4-8.

**Table 4-8: Three groups and subgroups based on their performances in the first comparison**

<table>
<thead>
<tr>
<th>Group</th>
<th>Sub group</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1a</td>
<td>Lift, and information gain</td>
</tr>
<tr>
<td></td>
<td>1b</td>
<td>One-way support and added value</td>
</tr>
<tr>
<td>2</td>
<td>2a</td>
<td>Confidence, example and counter example, Ganascia index, and Sebag-Schoenauer</td>
</tr>
<tr>
<td></td>
<td>2b</td>
<td>Certainty factor, conviction, odd multiplier, and Zhang</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Odd ratio, Yule’s Q, and Yule’s Y</td>
</tr>
</tbody>
</table>

From Table 4-1 and Table 4-8, the formula from all the criteria from group 1 and group 2 can be rewritten by involving only confidence (c) and probability of result P(B) but not support (s). Therefore, they cause problems for ties to a greater extent than do the other criteria that involve all of s, c, and P(B) in their formulas.

The first group contains four criteria that need the same number of rules to find the true model. The four criteria are lift, information gain, one-way support, and added value. Refer to Table 4-1, these four criteria can be re-written with only confidence (c) and probability of result (P(B)) without support (s) involved. Each of the four criteria monotonically increases with c but monotonically decreases with P(B). These four criteria can be divided further into two subgroups. Subgroup 1a, lift and information gain, always behave like each other as information gain is the logarithm form of lift. In other words, information gain always increases when lift increases, i.e., information monotonically increases with lift. Subgroup 1b behaves like subgroup 1a for top rules but not necessarily for the low-ranked rules. As we consider the number of rules needed for the true model, the top-
ranked rules play an important part in the comparison but the low-ranked rules do not affect the similarity among these criteria.

The second group contains eight criteria that need the same number of rules to find the true model. The eight criteria are confidence, example and counter example, Ganascia index, Sebag-Schoenauer, certainty factor, conviction, odd multiplier, and Zhang. These four criteria can be divided further into two subgroups. Subgroup 2a, confidence, example and counter example, Ganascia index, and Sebag-Schoenauer, can be re-written using only confidence (c) without both P(B) and support (s). They always behave like each other, as all the criteria in this subgroup always increase when confidence increases, i.e., they monotonically increase with confidence. Subgroup 2b behaves like subgroup 1a when confidence is equal to 1, which happens when $P(AB') = 0$. When this occurs, the four criteria in subgroup 2a give the maximum value. Note that when $c = 1$, the certainty factor becomes 1, conviction, and odd multiplier and Zhang become infinity. As we consider the number of rules needed for the true model, the top-ranked rules play an important part in the comparison but the low-ranked rules do not affect the similarity among these criteria.

For the third group, the three criteria are odd ratio, Yule’s Q, and Yule’s Y. They give the maximum value when confidence is equal to 1. Note that when this occurs, the odd ratio will become infinity and both Yule’s Q and Yule’s Y will become one. Different from group 2b, their values are maximum not only when $P(AB') = 0$ but also when $P(A'B) = 0$. Therefore, their performance is a little different from group 2.

**4.3.2 Comparison of the Effects of Increases in the Variables**

The second comparison is performed to find out which criteria depend on the number of variables more than the others. In other words, our goal is to determine when the models are the same how the number of rules needed change for each criterion when we increase the number of variables. Therefore, we compare the performance for each criterion with 6 and 10 binary variables. The performance comparisons are illustrated with bar charts in Appendix B and summarized in Table 4-9.

<table>
<thead>
<tr>
<th>scenario</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>accur</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>added</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>certa</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>conf</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>convi</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>coole</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cosin</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dice</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>direc</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exampl</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ganas</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gimi</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>impli</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>infor</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>jacca</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>jmeas</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kappa</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kloq</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kulpz</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lapla</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>least</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lever</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>lift</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>norma</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>oddmu</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>oddra</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>onesu</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PS</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>recal</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>relat</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>roger</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sebag</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>speci</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>taub</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>twosu</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>twova</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>yuleq</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>yuley</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zhang</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: ‘Y’ represents ‘yes’ meaning that there is a change in number of rules needed to find the true model between the two design matrices.

In this section, we need to give some definitions and properties to provide the basis of our analysis for this comparison.

**Definition:** The model that cannot be reduced to a simpler form is called the dominant model (refer to section 3.2.1). Hence, all the models and their complement models that appear in Table 4-3 are dominant models for class 1 and class 0, respectively. Note that in addition the complement models cannot be reduced to a simpler form for result $Y = 0$. All the rules in the dominant models are called dominant rules.
**Definition:** A child rule is the rule whose condition is the subset of its parent rule when the result is the same. As an example, the rule if $X_1 = 0$ and $X_2 = 0$ then $Y = 1$ is the child rule of the rule if $X_1 = 0$ then $Y = 1$.

**Definition:** The model is balanced when all the rules in the model have the same rule size. Refer to Table 4-3: the balanced models are Models 3, 6, 7, and 8 and the complement models for Model 1, 2, 3, 5 and 8.

**Definition:** If the values of $s$ and $P(B)$ become as and $aP(B)$ when $a > 1$ and the value of this criterion is still the same, this condition is denoted by $\text{cri}(as, aP(B)) = \text{cri}(s, P(B))$.

**Definition:** A criterion is considered monotone in confidence if it monotonically increases with confidence when support and $P(B)$ are fixed, as denoted by $\uparrow c$. A criterion is considered monotone in support if it monotonically increases with support when confidence and $P(B)$ are fixed, as denoted by $\uparrow s$. The criterion is considered anti-monotone in $P(B)$ if it monotonically decreases with $P(B)$ when confidence and $P(B)$ are fixed, as denoted by $\downarrow P(B)$.

For example, consider $\text{cosine} = \frac{sc}{\sqrt{P(B)}}$. This criterion will increase with $c$ when $s$ and $P(B)$ are fixed, will increase with $s$ when $c$ and $P(B)$ are fixed, and will decrease with $P(B)$ when $s$ and $c$ are fixed. Therefore, the cosine holds $\uparrow c \uparrow s \downarrow P(B)$. The partial derivative can be used to prove monotonicity as well.

Table 4-10 classifies the criteria on the basis of how they depend on these three conditions.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\uparrow s$</td>
<td>Support</td>
</tr>
<tr>
<td>$\uparrow s \downarrow P(B)$</td>
<td>Recall</td>
</tr>
<tr>
<td>$\uparrow c$</td>
<td>Confidence, Example and counter example, Ganascia index, Laplace and Sebag-Schoenauer</td>
</tr>
<tr>
<td>$\uparrow c \downarrow P(B)$</td>
<td>Added value, Certainty factor, Conviction, Information gain, Lift, Odd multiplier, One-way support, and Zhang</td>
</tr>
<tr>
<td>$\uparrow c \uparrow s \downarrow P(B)$</td>
<td>Cosine, Dice index, Directed contribution to Chi-squared, Implication index, Jaccard, Klosgen, Kulczynski index, Least contradiction, Piatetshy-Shapiro, Two-way support</td>
</tr>
<tr>
<td>$\uparrow c \downarrow s \downarrow P(B)$</td>
<td>Leverage</td>
</tr>
<tr>
<td>$\uparrow c \downarrow P(B)$</td>
<td>Accuracy, Collective Strength, Gini index, J-measure, Kappa, Normalized</td>
</tr>
</tbody>
</table>

Table 4-10: Criteria based on three conditions
For the last group, they $\uparrow$s with some conditions, e.g., for $c > 0.5$ or $c > P(B)$. However, these conditions are always satisfied for the rules that play a role in finding the number of rules needed in our study.

**Property 4-1**: Any rule that appears in Table 4-3 will rank better than its child rule for any criteria that holds $\uparrow$s.

**Proof** Any rule that appears in Table 4-3 has confidence = 1. Its child rule always has confidence = 1 with the same P(B) but less support ($s$). Therefore, the child rule of this rule always has less value for any criterion that holds $\uparrow$s by definition. In consequence, the rank of the child rule is always worse than the parent rule as well.

From Table 4-9, leverage is the only criterion for which a different number of rules is needed for all models. Note that among all the criteria, leverage is the only one that is anti-monotone with support (monotone decreases when support increases when confidence and P(B) are fixed) for all the possible values of confidence and for which P(B) or $\downarrow$s. Therefore, this criterion always chooses child rules over parent rules. This is because there are a large number of child rules for which ties cannot be broken and the number increases with the number of variables. Therefore, when the number of variables increases, this criterion is always affected. This is also the reason why the leverage performance is poor compared to the other criteria as well.

In cases when the criterion is monotone as in $s$, $\uparrow$s, the dominant rule always ranks better than its child rule, as they all have confidence = 1 with the same P(B). In consequence, the number of rules needed for this criterion is the smaller of the number needed to find the set of the dominant rules for the model and the number needed to find the set of the dominant rules for the complement model. This is different from the case in which leverage ranks the child rules before its dominant rules.

For Models 3 and 8, with the exception of leverage, no criteria are affected when the number of variables increases. The character of these two models is that they are balanced models and so their complement models (Model 3' and Model 8') are also balanced. The rule size is equal for Model 3 and its complement model (Model 3'), which is equal to 2. None of the criteria except leverage are affected by the increase in the child rules of these dominant
rules. Criteria that depend on $P(B)$ or to be specific $\downarrow P(B)$, choose dominant rules from the complement model over the dominant rules from Model 3 itself. Note that for Model 3, $P(Y = 1) = 0.4375$ and $P(Y = 0) = 0.5625$. Child rules from additional variables will not affect the ranks of these two rules from the complement model. The criteria that are not in the function $P(B)$ will rank all six dominant rules in the top rank that are tied together, and their ranks will not be affected by the increase in variables either. Recall that we use both support and rule size to break ties for this analysis.

The rule size for Model 8 is 3, whereas the rule size for the complement model (Model 8') is 2. It does not matter which side of the model between Model 8 or the complement model (Model 8') is ranked better, additional child rules from these dominant rules will not affect the ranking as they will have lower support and so be ranked lower than the dominant rule. Therefore, if the model is balanced and its complement model is also balanced, with the exception of leverage, none of the criteria will be affected in terms of performance.

For Models 1, 2, 6, and 7, the criteria that are affected are in group 2 and group 3 from Table 4-8, Lapalace and leverage. The reason why leverage is affected has already been explained. The similarity between the criteria in group 2 and group 3 is explained in section 4.3.1. For Laplace, the larger the number of instances, the closer the behavior to confidence. Therefore, the performance of Laplace when there are 1,024 instances can be explained in the same way as we explained the performance of confidence.

The similarity of the Models 1, 2, 6, and 7 is that between these models and their complement models (Models 1', 2', 6', and 7'), one model is balanced but the other is unbalanced. Moreover, an unbalanced model has a smaller rule size than the rules in a balanced model. Use confidence as a representative for this set of criteria and use Model 1 as an example: we have Model 1 as an unbalanced model and its complement model (Model 1') is balanced. The first rule (rule 1a) in Model 1, if $X_1 = 0$ then $Y = 1$, has rule size = 1, which is smaller than the rule size in its complement model, which is 2. When the number of independent variables increases from 6 to 10 variables, there will be more child rules for this first rule (rule 1a), which will add to the number of rules needed to find the true model. For example, when the number of variables is 10, the rule if $X_1 = 0$ and $X_7 = 0$ then $Y = 1$ is the additional child rule of the first rule (rule 1a), which has confidence = 1 with support = 0.25, which, in turn, is equal to the all the other dominant rules: rule 1b (if $X_2 = 0$ and $X_3 = 0$ then $Y = 1$), rule 1'a (if $X_1 = 1$ and $X_2 = 1$ then $Y = 0$), and rule 1'b (if $X_1 = 1$ and $X_3 = 1$ then $Y$
In consequence, the number of ties will increase and the number of the rules needed will increase as well. Note that confidence prefers the rules with higher confidence and higher support when support is used to break ties; therefore, the number of rules needed will be the best number between finding both rules in Model 1 or both rules in its complement model (Model 1'). Please also note that this set of criteria also suffers from finding the low support rules.

The other criteria are not affected, as they choose the two rules that comprise the balanced model in their top rules. Note that the two balanced rules are from a class with a lower P(B) and the other criteria follow ↓P(B).

For Model 4, all the criteria are affected, as this model and its complement model are unbalanced. When the number of variables increases from 6 to 10 variables, the additional child rules from the first rule of Model 4 and the additional child rules from the first rule of the complement model will add to the number of rules needed no matter what combination of models are ranked better for each criterion.

For Model 5, the model itself is not balanced but its complement model is balanced with the rule size = 2. Consider the second rule (rule 5b) of Model 5 with confidence (c_{5b}) = 1, support (s_{5b}) = 0.125, and P(B)_{5b} = P(Y=1) = 0.34375 and all the rules in the complement model with confidence(c') = 1 and support (s') = 0.25 and P(B)' = P(Y=0) = 0.65625. For this model, P(B)' / P(B)_{5b} = 1.91, whereas s' / s_{5b} = 2. There is a tradeoff between the changes in support and P(B) with close ratios when confidence is equal to 1. The criteria that rank rule 5b after the rules in the complement model 5 will not be affected by the additional child rules of rule 5a when the number of variables increases. The criteria that are not affected are those in group 2 and group 3 of Table 5: cosine, implication index, Kulczynski index, least contradiction, and Piatetshy-Shapiro. Use confidence to represent the criteria in group 2 and group 3: This criterion will rank rules in the complement set better since they have the same confidence with the higher support. For the other five criteria, we find that three of them (cosine, least contradiction, and Kulczynski index) satisfy the condition that cri(as, aP(B)) = cri(s, P(B)) when confidence = 1. Let a = 2, it can be seen that cri(2s, 1.9P(B)) > cri(2, 2P(B)) = cri(s, P(B)) as all the criteria are ↓P(B). The other two criteria (implication index and PS) hold this condition for some range of a, but not all a > 1. However, for this model, they both have a higher value for the rule in the complement set than for rule 5a. For group 1 in Table 4-8, as they do not have support in their formula, they have a disadvantage in this scenario.
This is because they choose rule 5b over the complement rules, as \( P(Y = 1) \) is less than \( P(Y = 1) \) in this case.

### 4.3.3 Results Summary

In all eight models, each model is characterized based on the whether the given models and the complement models are balanced or unbalanced. Leverage is sensitive to all the models, as it is \( \downarrow s \). All the criteria are affected when neither the given model nor its complement model is balanced. In addition, the criteria in group 2 and group 3 in Table 4-8 are sensitive to the scenario in which either given model or the complement model is unbalanced and there is a rule whereby an unbalanced model has a smaller rule size than does the balanced model. The child rule of the smaller size rule will add to the number of rules needed when the number of variables increases. The criteria in group 1 in Table 4-8 is sensitive to the scenario in which either given model or the complement model is unbalanced and where the unbalanced model has a lower \( P(B) \) than the balanced model does. The model that requires almost the same amount of tradeoff between the \( P(B) \) and support \( (s) \) between the rules of the unbalanced model and those of the balanced model causes difficulties for criteria includes accuracy, collective strength, directed contribution to Chi-square, Gini index, J-measure, Kappa, Klosgen, normalized mutual information, relative risk, Roger-Tanimoto index, tau-b, two-way support, and two-way support variation. In this study, five criteria are less sensitive than the others to changes in the variables; these are cosine, implication index, least contradiction, Kulczynski index, and PS. Note that these five criteria all follow \( \uparrow c \uparrow s \downarrow P(B) \).

In addition to our focus on the behavior of the criteria when the number of variables increases, we also observe the criteria that perform well and are robust with the scenarios of interest. To be specific, we look for criteria that rank the dominant models in their top ranks in all the models. As the dominant models are the rules with high confidence, the criteria that behave well in our study prefer the rules with high confidence values over the rules with high support values when there is a tradeoff between these two measurements.

Consider cosine, this criterion will be the same if support decreases in the same proportion that confidence increases, i.e., \( \text{cosine}(s,c) = \text{cosine}(s/a, ac) \) when \( P(B) \) is fixed and \( a > 1 \). This criterion is robust in all cases, with the exception of the poor performance in Model 4. Use the criterion of cosine as the reference, the criteria that behave better than cosine for each scenario are those that give more importance to a higher value of confidence.
than to a the higher value of support; i.e., cri(s,c) < cosine(s/a, ac), given P(B) is fixed and a > 1. With the value of P(B) given in all eight models, the criteria that outperform cosine include implication index, Kulczynski index, and least contradiction.

Note that Dice index, Jaccard, and specificity perform poorly for Model 4, as they choose the higher support rules over the higher confidence rules; i.e., cosine(s,c) > cosine(s/a, ac), when P(B) is fixed. For 'recall', performance is poor in almost all the models as this criterion does not take confidence into consideration.

4.3.4 Results for Simulations with Noise

In this part, we test the five criteria that are least sensitive to changes in the variables which are cosine, implication index, least contradiction, Kulczynski index, and PS. We add 10% noise to all eight models in Table 4-3 and test the models with rule selection criteria for 500 iterations. The simulation results are shown in histograms in Appendix D.

From the simulation results, Model 2 produces a greater difference in results between the circumstance with noise and the circumstance without noise than do the other models. Noise affects all the illustrated criteria for Model 2. The reason is that the given model is the most unbalanced compared to the other models (one rule size is 1 and the other rule size is 3). The child rule of rule 2a when noise is involved is in the better rank than rules 2'a, 2'b, and 2'c. If we use the median as the representative of the number of rules needed when noise is involved, the comparisons are as follows. The results are the same or very close for all the criteria in Model 1, Model 3, Model 4, and Model 7. For Model 5, the results are worse when noise is involved but with less effect than for Model 2. The reason for this model is that there are many rules for which the values are close without noise. When noise is involved, however, the ranking can be changed more easily for Model 5 than for the other models with the exception of Model 2. For Model 6 and Model 8, the effect of noise is very small for all the criteria except PS. Note that PS also shows higher variations than the other criteria in most cases.
4.4 Conclusions

One problem that distinguishes between the criteria is the tie problem. We study the dependency of the criteria on support and rule size to break ties. The result follows our expectation that criteria that can be rewritten in terms of support, confidence, and probability of the result $P(B)$ together do not need support and rule size to break ties. On the other hand, the criteria that do not contain support in their formula need support and rule size to break ties.

The effect of the number of variables is one of our concerns. We discovered that the character of each model plays an important role in the performance of each criterion. The character that we focus on is the balanced vs. unbalanced models between the given model and the complement models. In cases in which the model and its complement model are both balanced, the number of rules needed is the same when the number of variables increases for all criteria except leverage. In cases in which either the given model or its complement model is balanced, the number of rules needed will differ for the rules selection criteria that prefer the unbalanced model over the balanced model with the first design matrix 2(6). The reason is that when the number of variables increases, a problem arises in the unbalanced model as the additional child rule of the rule with a smaller size will add to the number of rules needed to find the true model. In cases in which both the given model and its complement model are unbalanced, all the rules selection criteria are affected.

We recommend the criteria that contain all three measurements, support, confidence, and $P(B)$ in their formula, and also monotone with confidence, monotone with support, and anti-monotone with $P(B)$. Moreover, another advantageous feature is for a criterion to give more importance to a higher confidence value than to a higher support value when there is a tradeoff, when $P(B)$ is fixed. These criteria will choose the dominant rules in the better rank compared to the criteria that prefer high support over high confidence when $P(B)$ is fixed.

This study of the effect of noise shows that some criteria are more sensitive to noise than other criteria are. Also, there may be some models, especially those that are highly unbalanced, that are more sensitive to noise than other models are. However, the selected criteria still perform well when noise is included in the models.
Chapter 5

Multinomial Logit Models Selection via Association Rules Analysis

In this chapter, we propose a model selection procedure for the multinomial logit model by implementing association rules analysis. We intend to demonstrate that the proposed method is practical for applying to other categorical models. The multinomial logit model is an enhanced version of the logistic regression model with more levels of response. The classical logistic regression model only considers main effects. It ignores interactions. When there are interactions in the models, the classical model loses an opportunity to capture these meaningful relationships among variables.

The multinomial logit models have been used widely for many decades. However, the same problem with the logistic regression also applies to multinomial logistic regression. The interactions among variables are usually omitted from the model selection process because considering them renders the process too complexity. It is even more complicated to involve interactions to the multi-level response model than in the binary response model. Therefore, in this study, we study the possible benefits of including these interactions in the multinomial logistic regression model and propose a framework that effectively selects significant interactions for the multinomial logistic regression model.

This chapter is organized as follows. Section 5.1 offers a review of the relevant academic literature pertaining to multinomial logistic regression modeling and association rules analysis, and Section 5.2 presents an application of the proposed method with a simulated dataset. Section 5.3 presents the application of our framework to a real dataset. Section 5.4 offers a discussion and concluding remarks.
5.1 Multinomial Logistic Regression Model and Association Rules Analysis

Logistic regression has been widely used in many academic areas and on the basis of many applications. It has been used to explore relationships among variables and to predict binary response classes based on explanatory variable values. And, multinomial logistic regression models have been developed for multiple-level responses. Based on Cramer (2003), early work includes studies by Gurland et al. (1960), Cox (1966), Theil (1969).

In this study, we focus on the multicategory response without orders or nominal data. The multinomial logit model for this type of response is the baseline-category logit model, which combines with a separate binary logit for each pair of response categories (Agresti 2002). The baseline-category logit model with all the main effects is illustrated here:

Let \( \pi_j(x) = P(Y = j|x) \) at a fixed setting of \( x \) and \( \sum_j \pi_j(x) = 1 \) (Agresti, 2002).

Logit models pair each response with a baseline category, often the last one or the most common one. The model

\[
\log \frac{\pi_j(x)}{\pi_{j'}(x)} = \alpha_j + \beta_j x, \quad j = 1, \ldots, J - 1
\]

simultaneously describes the effects of \( x \) on these \( J-1 \) logits.

Interactions can be addressed in different ways, especially between categorical variables. In the classical method, interactions are omitted from the logit models (Agresti 2002). In our study, we consider interactions and main effects simultaneously in the multinomial logit model. As the number of interactions increases at an accelerated rate with a higher number of main effects, the subset selection method is extremely inefficient when all the interactions are considered at the same time. Therefore, an efficient method for selecting potential interaction variables is required. We propose implementing association rules analysis.

To our knowledge, there are no studies linking association rules analysis and classification rules mining to logistic regression modeling. In our study, we apply classification rule mining to screen out the unnecessary interactions and keep only the potentially significant ones to be considered in building the multinomial logit model. This methodology is used as a major aspect of the variables selection in our process.
Some studies have developed other techniques for selecting and screening variables for the multinomial logit model. Zahid and Tutz (2010) use the likelihood-based boosting technique with one step of Fisher scoring in variable selection, whereas Cherrie (2008) developed the five-step technique that involves ANOVA and bootstrapping aggregation in variable screening. However, they did not consider the interactions among variables in the selection process. Lucadamo et al. (2010) uses principle component analysis to eliminate the problem of multicollinearity data, which can also reduce the number of variables in the multinomial logit model fitting. Kim and Kim (2010) developed methodology that combines a decision tree with the multinomial logit model. In their two-stage method, they use the decision tree at the beginning to select the influential main effects and interactions to act as the explanatory variables for multinomial logit model fitting in the last step. Adams and Wilson (1996) present a technique called the random coefficients multinomial logit model. In our proposed method, we use association rules as a part of global search for interactions for the multinomial logit model.

5.2 Proposed Method with the Multinomial Logit Model

The proposed framework for building a model that predicts a multiclass response from binary explanatory variables consists of four key steps, the same as the four steps shown in Figure 2-1 of Chapter 2. The work-flow of the proposed method is the same as that shown in Figure 2-2. However, in step 4, we use deviance as a criterion to select the optimal model. Moreover, in step 1, the minimum support needs adjustment as the minimum support of 10% seems to be too high for multi-level response. The level of 5% is recommended based on our experience with simulated datasets and real datasets. The minimum confidence should be adjusted accordingly as well.

For illustration and comparison, we modify the MONK’s dataset (the first dataset from the MONK’s problem) by giving more levels of response and adjust the criteria for the multiclass. The objective of this adaptation is to classify 432 robots into three classes (class 1, class 2, and class 3), based on six attributes. The details for all the attributes are the same as shown in Table 2-1. The true model will assign the robot into three classes with a subset of the six attributes. In this study, we use the proposed method to find the multinomial logit
model that fits this dataset and then to compare the results to those generated by the classical multinomial logistic regression technique.

The underlying (true) model is shown in Table 5-1 that classifies robots into class 1 \((Y = 1)\), class 2 \((Y = 2)\), or class 3 \((Y = 3)\). Note that the true model for the simulated dataset is designed by the following logic (the truth).

<table>
<thead>
<tr>
<th>Table 5-1 Variable class results for the dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 3 ((Y = 3))</td>
</tr>
<tr>
<td>(X_8 = 1)</td>
</tr>
<tr>
<td>(the robot wears a red jacket)</td>
</tr>
<tr>
<td>(X_1 = 1 ) and (X_3 = 1)</td>
</tr>
<tr>
<td>(the robot has a round head shape and a round body shape)</td>
</tr>
</tbody>
</table>

| Class 2 \((Y = 2)\)                              |
| \(X_1 = 1 \) and \(X_4 = 1\)                    |
| (the robot has a square head shape and a square body shape) |

| Class 1 \((Y = 1)\)                              |
| \(X_1 = 1, X_3 = 0 \) and \(X_8 = 0\)           |
| (the robot (i) has a round head shape, (ii) does not have a round body shape, and (iii) does not wear a red jacket) |

| Rule 1: If \(X_8 = 1\), \(Y = 3\) with \(s = 0.2500\), \(c = 1.000\); |
| Rule 2: If \(X_1 = 1, X_3 = 0, X_8 = 0\), then \(Y = 1\) with \(s = 0.1667\), \(c = 1.000\). |
| Rule 3: If \(X_1 = 0, X_3 = 1, X_8 = 0\), then \(Y = 1\) with \(s = 0.1667\), \(c = 1.000\). |
| Rule 4: If \(X_2 = 1, X_4 = 0, X_8 = 0\), then \(Y = 1\) with \(s = 0.1667\), \(c = 1.000\). |
| Rule 5: If \(X_2 = 0, X_4 = 1, X_8 = 0\), then \(Y = 1\) with \(s = 0.1667\), \(c = 1.000\). |

Step 1: Use CBA to find the association rules. Note that we use a minimum support value of 5% and a minimum confidence value of 80% to generate the active rules.

Step 2: Select the top 30 rules based on confidence criteria. Examples of the selected rules are:

| Rule 1: If \(X_8 = 1\), \(Y = 3\) with \(s = 0.2500\), \(c = 1.000\); |
| Rule 2: If \(X_1 = 1, X_3 = 0, X_8 = 0\), then \(Y = 1\) with \(s = 0.1667\), \(c = 1.000\). |
| Rule 3: If \(X_1 = 0, X_3 = 1, X_8 = 0\), then \(Y = 1\) with \(s = 0.1667\), \(c = 1.000\). |
| Rule 4: If \(X_2 = 1, X_4 = 0, X_8 = 0\), then \(Y = 1\) with \(s = 0.1667\), \(c = 1.000\). |
| Rule 5: If \(X_2 = 0, X_4 = 1, X_8 = 0\), then \(Y = 1\) with \(s = 0.1667\), \(c = 1.000\). |

Step 3: Convert the 30 rules into 30 variables. For example, rule 2 was converted into the new variable called \(X_1(1)X_3(0)X_8(0)\), where
Step 4: Combine the 30 variables with the main effects and use the subset selection method to search for the optimal baseline category logit model that yields the optimal deviance.

The optimal baseline category logit model is the model that contains four variables as shown below:

\[
\ln \left( \frac{p_1}{p_3} \right) = 15.0488 - 29.3586X_8 - 14.7229X_1(1)X_3(1) - 14.7229X_2(1)X_4(1) \\
- 14.7229X_1(0)X_2(0)X_3(0)X_4(0).
\]

\[
\ln \left( \frac{p_2}{p_3} \right) = 0.5357 - 29.3586X_8 - 14.7229X_1(1)X_3(1) - 14.7229X_2(1)X_4(1) \\
- 14.7229X_1(0)X_2(0)X_3(0)X_4(0).
\]

According to the estimated coefficients from the four variables, the model can be explained using four conditions according to each variable:

i. If \( X_8 = 1 \) then \( Y = 3 \).

ii. If \( X_1(1)X_3(1) = 1 \), meaning \( X_1 = 1 \) and \( X_3 = 1 \), then \( Y = 2 \).

iii. If \( X_2(1)X_4(1) = 1 \), meaning \( X_2 = 1 \) and \( X_4 = 1 \), then \( Y = 2 \).

iv. If \( X_1(0)X_2(0)X_3(0)X_4(0) = 1 \), meaning \( X_1 = 0 \), \( X_2 = 0 \), \( X_3 = 0 \) and \( X_4 = 0 \), then \( Y = 2 \).

These four conditions are exactly the same conditions as those shown in Table 5-1. Therefore, this model can capture the whole logic behind the true model.

The optimal classical baseline category logit model is as shown below:

\[
\ln \left( \frac{p_1}{p_3} \right) = 26.9347 - 54.9717X_8
\]

\[
\ln \left( \frac{p_2}{p_3} \right) = 26.2416 - 54.2785X_8
\]

Adding more variables when \( X_8 \) is already included in the model does not improve the deviance, as the parameters for the other variables are equal to zero. Based on this model, the classical method only captures the significance of the color of the jacket (\( X_8 \)), and does not capture the significance of the robot’s head shape or body shape (\( X_1, X_2, X_3, \) and \( X_4 \)). According to this variable, if \( X_8 = 1 \) then \( Y = 3 \).
The modified MONK’s dataset adequately illustrates our proposed method and demonstrates its ability to capture the interactions between variables. Any method that does not take high-order interactions (three-order interactions and four-order interactions) into consideration would not be able to obtain the true model for this dataset. We illustrate the implementation of the proposed method based on a real-life dataset in the next section.

5.3 Application: Alligator Food Choice Dataset

In this section, we use an alligator food choice dataset from Agresti (2002). As mentioned in Agresti’s book, this data is courtesy of Clint Moore, from an unpublished study by M. F. Delaney and C. T. Moore. Here, we illustrate an application of our proposed method by selecting the model for this dataset using our framework.

The alligator food choice dataset (Agresti 2002) is used to study factors that influence the primary food choice of alligators. There are five types: fish, invertebrate, reptile, bird, and other. There are three original categorical variables: lake, gender, and size. We converted all the responses and attributes into binary variables, as listed in Table 5-2.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Levels</th>
<th>Binary Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary food choice</td>
<td>fish, invertebrate, reptile, bird, and other</td>
<td>Y = 1 if the class is invertebrate&lt;br&gt;Y = 2 if the class is reptile&lt;br&gt;Y = 3 if the class is bird&lt;br&gt;Y = 4 if the class is other&lt;br&gt;Y = 5 if the class is fish</td>
</tr>
<tr>
<td>Lake</td>
<td>Hancock, Oklawaha, Trafford, George</td>
<td>X1 = 1 if Hancock and X1 = 0, otherwise&lt;br&gt;X2 = 1 if Oklawaha and X2 = 0, otherwise&lt;br&gt;X3 = 1 if Trafford and X3 = 0, otherwise</td>
</tr>
<tr>
<td>Gender</td>
<td>male and female</td>
<td>X4 = 1 if male and X4 = 0, otherwise</td>
</tr>
<tr>
<td>Size (m)</td>
<td>≤ 2.3 and &gt; 2.3</td>
<td>X5 = 1 if ≤ 2.3 and X5 = 0, otherwise</td>
</tr>
</tbody>
</table>

Note: We use fish as the baseline category (class 5).

We applied the proposed method to this dataset and obtained the following results:

**Step 1:** Used CBA to obtain the active rules. Note that we use a minimum support value of 5% and omit the minimum confidence, as the highest confidence for the rules with more than 5% minimum support is 77.27%.

**Step 2:** Selected the 30 rules with the highest confidence values from among all the active rules. Examples of the selected rules follow:

Rule 1: If X1 = 0, X2 = 0, X3 = 0, X5 = 0 then Y = 5 with s = 0.0776, c = 0.7727.
Rule 2: If $X_1 = 1$, $X_4 = 0$, $X_5 = 1$ then $Y = 5$ with $s = 0.0731$, $c = 0.6154$.

Rule 3: If $X_1 = 0$, $X_3 = 0$, $X_4 = 0$, $X_5 = 1$ then $Y = 1$ with $s = 0.0822$, $c = 0.6207$.

Rule 4: If $X_2 = 0$, $X_3 = 0$, $X_4 = 1$, $X_5 = 0$ then $Y = 5$ with $s = 0.0594$, $c = 0.6842$.

Rule 5: If $X_2 = 1$, $X_5 = 1$ then $Y = 1$ with $s = 0.0502$, $c = 0.5500$.

**Step 3:** Converted the 30 selected rules into variables. For example, rule 1 was converted into the new variable called $X_1(0)X_2(0)X_3(0)X_5(0)$, where

$$X_1(0)X_2(0)X_5(0)X_5(0) = \begin{cases} 1 & \text{if } X_1 = 0, X_2 = 0, X_3 = 0, X_5 = 0 \\ 0 & \text{otherwise} \end{cases}$$

**Step 4:** Combined the 30 variables with the main effects and used the subset selection method to search for the optimal baseline category logit model.

Please note that we would like to compare the model from the proposed method with the model from the classical method. From the classical model, the optimal model in terms of deviance is the full model with five variables with a deviance of 537.8655. However, the deviance of the five-variable model from the proposed method is 522.6545. Therefore, the proposed method can improve on the goodness of fit of the classical method.

The baseline category logit model from the proposed method with five variables is as shown below:

$$\ln \left( \frac{p_1}{p_5} \right) = -0.9741 + 13.5729X_1(0)X_2(0)X_4(0)X_5(0) - 15.2166X_2(0)X_3(0)X_5(0)$$

$$+ 0.5383X_1(0)X_2(0)X_5(0) - 0.9731X_2(0)X_3(0)X_4(1) + 1.7619X_1(0)X_5(1)$$

$$\ln \left( \frac{p_2}{p_5} \right) = -1.0717 - 15.3623X_1(0)X_2(0)X_4(0)X_5(0) - 0.0559X_2(0)X_3(0)X_5(0)$$

$$+ 0.3483X_1(0)X_3(0)X_5(0) - 14.0207X_2(0)X_3(0)X_4(1) - 0.1939X_1(0)X_5(1)$$

$$\ln \left( \frac{p_3}{p_5} \right) = -1.6908 - 14.5898X_1(0)X_2(0)X_4(0)X_5(0) + 1.1470X_2(0)X_3(0)X_5(0)$$

$$- 0.9893X_1(0)X_3(0)X_5(0) - 0.5903X_2(0)X_3(0)X_4(1) - 0.2422X_1(0)X_5(1)$$

$$\ln \left( \frac{p_4}{p_5} \right) = -0.8987 - 0.2288X_1(0)X_2(0)X_4(0)X_5(0) + 0.9088X_2(0)X_3(0)X_5(0)$$

$$- 1.8607X_1(0)X_3(0)X_5(0) - 0.2183X_2(0)X_3(0)X_4(1) + 0.1319X_1(0)X_5(1)$$

The probability of being in each class can be estimated from the model. However,
according to the estimated coefficients from the five variables, the model can be explained using five conditions according to each variable.

i. If Lake is neither Hancock nor Oklawaha, Gender is female, size > 2.3m, then the probability that the primary food choice is fish is always higher than the probability that the primary food choice is reptile, bird, or other.

ii. If Lake is neither Oklawaha nor Trafford, size > 2.3m, then the probability that the primary food choice is fish is always higher than the probability that the primary food choice is invertebrate or reptile.

iii. If Lake is neither Hancock nor Trafford, Gender is female, size > 2.3m, then the probability that the primary food choice is fish is always higher than the probability that the primary food choice is bird or other.

iv. If Lake is not either Oklawaha or Trafford, Gender is male, then the probability that the primary food choice is fish is always higher than the probability that the primary food choice is reptile, bird, or other.

v. If Lake is not Hancock, size ≤ 2.3m, then the probability that the primary food choice is fish is always higher than the probability that the primary food choice is reptile or bird.

5.4 Conclusions

Through the demonstration in this section, the proposed method is easily applied to the multinomial logit model and the results show the effectiveness of the proposed method with the model when it is applied to logistic regression model. However, the suggested thresholds (minimum support and minimum confidence) for the proposed method for the multinomial logit model can be different from the logistic regression model. We realize that the study of the compatibility between the proposed method and the multinomial logit models and other types of categorical models are still in the initial stages. Arbitrariness may need to be adjusted and other types of adjustment may also be required, and these are factors that I will consider in my future work.
Chapter 6

Conclusion and Future Work

A central problem in logistic regression modeling is accounting for interaction effects, specifically those between categorical variables. In this research, we propose a framework that is capable of selecting potential interactions from a large number of candidates and including those selections in the variable selection process. Typically neglected in logistic model building, significant higher-order interactions can be selected and incorporated into the model. Our study confirms that the methodology proposed herein is effective in selecting interactions that improve model fit and facilitate our understanding of datasets. The popular model selection methods in logistic regression do not work well in general, as indicated in the simulation.

There is some arbitrariness in the proposed method, e.g., the minimum support in step 1 and the number of selected rules in step 2. The determination of the cut-off points has always been an issue for association rules analysis. The minimum thresholds always depend on practitioners. We propose a model-building procedure based on the commonly used threshold, which also works well with our dataset. The simulations also confirm that the recommended threshold works well with general datasets with the involvement of interactions. Note that the recommended number of rules also needed to be confirmed by the simulation results in Chapter 5.

Association rules analysis typically considers a given model as the true model and ignores all other possibilities. The idea of models that generate the same dataset has never been adequately considered. The problem of such an equivalency occurs for any dataset. Moreover, this problem leads to misleading results in terms of methodology development and comparison.

This research explores the issue of equivalent models thoroughly. Not only do we focus on preventing misleading results, but we also provide the solution for unstable explanations from equivalent models. We propose using the benchmark model (the dominant model) as the representative for explaining the relationship between variables. The derivative of the benchmark model is provided. By using a well-known dataset, we emphasize that equivalence is sensitive to previous studies in association rules analysis. We hope that this
discovery will provide a new aspect not only for association rules analysis but also for other related data analysis methodologies.

As an important part of rules selection criteria in association rules analysis, we analyze the behavior of rules selection criteria through a methodically designed framework in order to search for criteria that are robust for finding the true models from the scenarios of interest. The objective was to analyze the performance of criteria fairly supported by evidence from systematic experiments. The analysis was based on eight models, which varied in terms of character and the interactions among the variables. However, to compare the rules section criteria fairly, the issue of equivalent models in association rules analysis needs to be given serious consideration.

Two principal problems that generally occur in association rules analysis are those of breaking a tie and considering a large number of variables. We study the effect of these two problems with each criterion and recommend the properties of rules selection criteria that are most practical for negotiating them. For the tie problem, the selections of criteria that are more practical than the others are those with formulas that include support \( (s) \). For an effect of number of variables, we discovered that the character of each model plays an important role in the performance of each criterion. The character that we focus on is the unbalanced vs. unbalanced models between the given model and the complement model. Without understanding this character, the analysis for criteria comparison would not be complete and the understanding of the behavior for each criterion would not be fulfilled. With these considerations, we were able to recommend criteria that are practical for the scenarios of interest. We recommend the criteria that are monotone in confidence, monotone in support and anti-monotone in probability of result \( P(B) \) and prefer the rules with high confidence values over the rules with high support values when there is a tradeoff between these two measurements.

Our intention is to apply the proposed method with a variety of statistical models for categorical variables. Therefore, we applied the proposed method with the multinomial logit model and the result showed the effectiveness of the proposed method with the model as well as its applicability to the logistic regression model. However, we realize that the study of the compatibility between the proposed method and the multinomial logit models and other types of categorical models is still at an early stage. Some adjustments, including for arbitrariness, may be needed. This leads to a consideration of my future work.

My future work will include the following topics:
i) Further study of the adjustments that may be needed when applying the proposed method with the multinomial logit model including the thresholds for rules selection criteria and the number of rules needed for the rules selection stage. The study will be through simulation processes, which is similar to the procedures used in Chapter 4 but with different models and perhaps different design matrices.

ii) Study the proposed method’s compatibility with the other categorical models including the Poisson regression model. The process will begin with studying how compatible the proposed method is with a simulated dataset and then with a real dataset. Also, the disadvantages of the proposed method with the model of interest will be observed.

iii) Analyze the power of the proposed method with the selected rules selection criteria to find the true models with different levels of noise, different numbers of variables, and different numbers of rules needed with the recommended rules selection criteria.
Appendices

Appendix A: Bar charts for comparing the number of rules needed with and without the involvement of support and rule size to break ties

Appendix B: Bar charts for comparing the number of rules needed for input matrix with 6 and input matrix with 10 variables

Appendix C: Simulation results summary

Appendix D: Histograms for the number of rules needed for Model 1–Model 8 with 10% noise

Appendix E: Matlab Codes
Appendix A

Bar Charts for Comparing the Number of Rules Needed With and Without the Involvement of Support and Rule Size to Break Ties
Figure A1: Bar chart representing the number of rules needed for Model 1
Blue bar for without support and rule size and pink bar for with support and rule size to break ties

Figure A2: Bar chart representing the number of rules needed for Model 2
Blue bar for without support and rule size and pink bar for with support and rule size to break ties
Figure A3: Bar chart representing the number of rules needed for Model 3
Blue bar for without support and rule size and pink bar for with support and rule size to break ties

Figure A4: Bar chart representing the number of rules needed for Model 4
Blue bar for without support and rule size and pink bar for with support and rule size to break ties
Figure A5: Bar chart representing the number of rules needed for Model 5
Blue bar for without support and rule size and pink bar for with support and rule size to break ties

Figure A6: Bar chart representing the number of rules needed for Model 6
Blue bar for without support and rule size and pink bar for with support and rule size to break ties
Figure A7: Bar chart representing the number of rules needed for Model 7
Blue bar for without support and rule size and pink bar for with support and rule size to break ties

Figure A8: Bar chart representing the number of rules needed for Model 8
Blue bar for without support and rule size and pink bar for with support and rule size to break ties
Appendix B

Bar charts for comparing the number of rules needed for
input matrix with 6 binary variables and input matrix with 10 binary variables
**Figure B1:** Bar chart representing the number of rules needed for Model 1
Blue bar for 6 variables and pink bar for 10 variables (Results for support and leverage are not included)

**Figure B2:** Bar chart representing the number of rules needed for Model 2
Blue bar for 6 variables and pink bar for 10 variables (Results for support and leverage are not included)
Figure B3: Bar chart representing the number of rules needed for Model 3
Blue bar for 6 variables and pink bar for 10 variables (Results for support and leverage are not included)

Figure B4: Bar chart representing the number of rules needed for Model 4
Blue bar for 6 variables and Pink bar for 10 variables (Results for recall, support, and leverage are not included)
Figure B5: Bar chart representing the number of rules needed for Model 5
Blue bar for 6 variables and pink bar for 10 variables (Results for recall and leverage are not included)

Figure B6: Bar chart representing the number of rules needed for Model 6
Blue bar for 6 variables and pink bar for 10 variables (Results for support are not included)
Figure B7: Bar chart representing the number of rules needed for Model 7
Blue bar for 6 variables and pink bar for 10 variables (Results for recall and leverage are not included)

Figure B8: Bar chart representing the number of rules needed for Model 8
Blue bar for 6 variables and pink bar for 10 variables (Results for recall and leverage are not included)
Appendix C

Simulation Results Summary
Table C1: Number of rules needed for the design matrix 2(6) using scheme 1

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
<th>Model 5</th>
<th>Model 6</th>
<th>Model 7</th>
<th>Model 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>accur</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>14</td>
<td>10</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>added</td>
<td>59</td>
<td>79</td>
<td>65</td>
<td>39</td>
<td>40</td>
<td>13</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>certa</td>
<td>216</td>
<td>216</td>
<td>170</td>
<td>216</td>
<td>180</td>
<td>262</td>
<td>226</td>
<td>200</td>
</tr>
<tr>
<td>conf</td>
<td>216</td>
<td>216</td>
<td>170</td>
<td>216</td>
<td>180</td>
<td>262</td>
<td>226</td>
<td>200</td>
</tr>
<tr>
<td>convi</td>
<td>216</td>
<td>216</td>
<td>170</td>
<td>216</td>
<td>180</td>
<td>262</td>
<td>226</td>
<td>200</td>
</tr>
<tr>
<td>coole</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>14</td>
<td>14</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>cosin</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>18</td>
<td>14</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>direc</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>22</td>
<td>16</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>examp</td>
<td>216</td>
<td>216</td>
<td>170</td>
<td>216</td>
<td>180</td>
<td>262</td>
<td>226</td>
<td>200</td>
</tr>
<tr>
<td>ganas</td>
<td>216</td>
<td>216</td>
<td>170</td>
<td>216</td>
<td>180</td>
<td>262</td>
<td>226</td>
<td>200</td>
</tr>
<tr>
<td>ginii</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>13</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>impli</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>11</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>infor</td>
<td>59</td>
<td>79</td>
<td>65</td>
<td>39</td>
<td>40</td>
<td>13</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>jaccsa</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>22</td>
<td>16</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>jmeas</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>11</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>kappa</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>14</td>
<td>14</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>klosq</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>kucz</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>13</td>
<td>9</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>lapla</td>
<td>14</td>
<td>14</td>
<td>6</td>
<td>14</td>
<td>7</td>
<td>22</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>least</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>15</td>
<td>12</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>lever</td>
<td>42</td>
<td>55</td>
<td>47</td>
<td>29</td>
<td>30</td>
<td>11</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>lift</td>
<td>59</td>
<td>79</td>
<td>65</td>
<td>39</td>
<td>40</td>
<td>13</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>norma</td>
<td>42</td>
<td>55</td>
<td>47</td>
<td>30</td>
<td>30</td>
<td>13</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>oddmu</td>
<td>216</td>
<td>216</td>
<td>170</td>
<td>216</td>
<td>180</td>
<td>262</td>
<td>226</td>
<td>200</td>
</tr>
<tr>
<td>oddra</td>
<td>217</td>
<td>217</td>
<td>170</td>
<td>217</td>
<td>180</td>
<td>265</td>
<td>227</td>
<td>200</td>
</tr>
<tr>
<td>onesu</td>
<td>59</td>
<td>79</td>
<td>65</td>
<td>39</td>
<td>40</td>
<td>13</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>PS</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>16</td>
<td>11</td>
<td>3</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>recal</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>47</td>
<td>37</td>
<td>11</td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td>relat</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>11</td>
<td>12</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>roger</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>14</td>
<td>10</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>sebag</td>
<td>216</td>
<td>216</td>
<td>170</td>
<td>216</td>
<td>180</td>
<td>262</td>
<td>226</td>
<td>200</td>
</tr>
<tr>
<td>speci</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>18</td>
<td>24</td>
<td>5</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>support</td>
<td>27</td>
<td>28</td>
<td>18</td>
<td>26</td>
<td>19</td>
<td>32</td>
<td>26</td>
<td>21</td>
</tr>
<tr>
<td>taub</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>12</td>
<td>10</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>twosu</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>14</td>
<td>12</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>twova</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>13</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>yuleq</td>
<td>217</td>
<td>217</td>
<td>170</td>
<td>217</td>
<td>180</td>
<td>265</td>
<td>227</td>
<td>200</td>
</tr>
<tr>
<td>yuley</td>
<td>217</td>
<td>217</td>
<td>170</td>
<td>217</td>
<td>180</td>
<td>265</td>
<td>227</td>
<td>200</td>
</tr>
<tr>
<td>zhang</td>
<td>216</td>
<td>216</td>
<td>170</td>
<td>216</td>
<td>180</td>
<td>262</td>
<td>226</td>
<td>200</td>
</tr>
</tbody>
</table>
Table C2: Number of rules needed for the design matrix 2(6) using scheme 2

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
<th>Model 5</th>
<th>Model 6</th>
<th>Model 7</th>
<th>Model 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accur</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>14</td>
<td>10</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Added</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Certa</td>
<td>14</td>
<td>14</td>
<td>6</td>
<td>14</td>
<td>7</td>
<td>22</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>Conf</td>
<td>14</td>
<td>14</td>
<td>6</td>
<td>14</td>
<td>7</td>
<td>22</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>Convi</td>
<td>14</td>
<td>14</td>
<td>6</td>
<td>14</td>
<td>7</td>
<td>22</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>Coole</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>14</td>
<td>14</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Cosin</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>18</td>
<td>14</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Dice</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>22</td>
<td>16</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Direc</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Examp</td>
<td>14</td>
<td>14</td>
<td>6</td>
<td>14</td>
<td>7</td>
<td>22</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>Ganas</td>
<td>14</td>
<td>14</td>
<td>6</td>
<td>14</td>
<td>7</td>
<td>22</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>Gini</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>13</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>impli</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>11</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Infor</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Jacca</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>22</td>
<td>16</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Jmeas</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>11</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Kappa</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>14</td>
<td>14</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Klosq</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Kulcz</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>13</td>
<td>9</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Lapla</td>
<td>14</td>
<td>14</td>
<td>6</td>
<td>14</td>
<td>7</td>
<td>22</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>Least</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>15</td>
<td>12</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Lever</td>
<td>42</td>
<td>55</td>
<td>47</td>
<td>29</td>
<td>30</td>
<td>11</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Lift</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Norma</td>
<td>42</td>
<td>55</td>
<td>47</td>
<td>30</td>
<td>30</td>
<td>13</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Oddmu</td>
<td>14</td>
<td>14</td>
<td>6</td>
<td>14</td>
<td>7</td>
<td>22</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>Oddra</td>
<td>15</td>
<td>15</td>
<td>6</td>
<td>15</td>
<td>7</td>
<td>22</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>Onesu</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>PS</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>16</td>
<td>11</td>
<td>3</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>Recal</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>47</td>
<td>37</td>
<td>11</td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td>Relat</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>11</td>
<td>12</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Roger</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>14</td>
<td>10</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>sebag</td>
<td>14</td>
<td>14</td>
<td>6</td>
<td>14</td>
<td>7</td>
<td>22</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>speci</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>18</td>
<td>24</td>
<td>5</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>support</td>
<td>27</td>
<td>28</td>
<td>18</td>
<td>26</td>
<td>19</td>
<td>32</td>
<td>26</td>
<td>21</td>
</tr>
<tr>
<td>taub</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>12</td>
<td>10</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>twosu</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>14</td>
<td>12</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>twova</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>13</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>yuleq</td>
<td>15</td>
<td>15</td>
<td>6</td>
<td>15</td>
<td>7</td>
<td>22</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>yuley</td>
<td>15</td>
<td>15</td>
<td>6</td>
<td>15</td>
<td>7</td>
<td>22</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>zhang</td>
<td>14</td>
<td>14</td>
<td>6</td>
<td>14</td>
<td>7</td>
<td>22</td>
<td>15</td>
<td>9</td>
</tr>
</tbody>
</table>
### Table C3: Number of rules needed for the design matrix 2(10) using scheme 2

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
<th>Model 5</th>
<th>Model 6</th>
<th>Model 7</th>
<th>Model 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>accur</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>22</td>
<td>18</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>added</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>18</td>
<td>18</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>certa</td>
<td>22</td>
<td>22</td>
<td>6</td>
<td>22</td>
<td>7</td>
<td>38</td>
<td>23</td>
<td>9</td>
</tr>
<tr>
<td>conf</td>
<td>22</td>
<td>22</td>
<td>6</td>
<td>22</td>
<td>7</td>
<td>38</td>
<td>23</td>
<td>9</td>
</tr>
<tr>
<td>convi</td>
<td>22</td>
<td>22</td>
<td>6</td>
<td>22</td>
<td>7</td>
<td>38</td>
<td>23</td>
<td>9</td>
</tr>
<tr>
<td>coole</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>22</td>
<td>22</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>cosin</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>26</td>
<td>14</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>dice</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>38</td>
<td>24</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>direc</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>18</td>
<td>18</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>examp</td>
<td>22</td>
<td>22</td>
<td>6</td>
<td>22</td>
<td>7</td>
<td>38</td>
<td>23</td>
<td>9</td>
</tr>
<tr>
<td>ganas</td>
<td>22</td>
<td>22</td>
<td>6</td>
<td>22</td>
<td>7</td>
<td>38</td>
<td>23</td>
<td>9</td>
</tr>
<tr>
<td>ginii</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>21</td>
<td>18</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>impli</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>19</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>infor</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>18</td>
<td>18</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>jacea</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>38</td>
<td>24</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>jmeas</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>19</td>
<td>18</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>kappa</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>22</td>
<td>22</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>klosq</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>18</td>
<td>18</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>kulcz</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>21</td>
<td>9</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>lapla</td>
<td>22</td>
<td>22</td>
<td>6</td>
<td>22</td>
<td>7</td>
<td>38</td>
<td>23</td>
<td>9</td>
</tr>
<tr>
<td>least</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>23</td>
<td>12</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>lever</td>
<td>210</td>
<td>295</td>
<td>223</td>
<td>125</td>
<td>126</td>
<td>27</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>lift</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>18</td>
<td>18</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>norma</td>
<td>210</td>
<td>295</td>
<td>223</td>
<td>126</td>
<td>126</td>
<td>29</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>oddmu</td>
<td>22</td>
<td>22</td>
<td>6</td>
<td>22</td>
<td>7</td>
<td>38</td>
<td>23</td>
<td>9</td>
</tr>
<tr>
<td>oddra</td>
<td>23</td>
<td>23</td>
<td>6</td>
<td>23</td>
<td>7</td>
<td>38</td>
<td>23</td>
<td>9</td>
</tr>
<tr>
<td>onesu</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>18</td>
<td>18</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>PS</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>24</td>
<td>11</td>
<td>3</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>recal</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>87</td>
<td>69</td>
<td>11</td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td>relat</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>19</td>
<td>20</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>roger</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>22</td>
<td>18</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>sebag</td>
<td>22</td>
<td>22</td>
<td>6</td>
<td>22</td>
<td>7</td>
<td>38</td>
<td>23</td>
<td>9</td>
</tr>
<tr>
<td>speci</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>34</td>
<td>32</td>
<td>5</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>support</td>
<td>43</td>
<td>44</td>
<td>26</td>
<td>42</td>
<td>27</td>
<td>56</td>
<td>42</td>
<td>29</td>
</tr>
<tr>
<td>taub</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>20</td>
<td>18</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>twosu</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>22</td>
<td>20</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>twova</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>21</td>
<td>18</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>yuleq</td>
<td>23</td>
<td>23</td>
<td>6</td>
<td>23</td>
<td>7</td>
<td>38</td>
<td>23</td>
<td>9</td>
</tr>
<tr>
<td>yuley</td>
<td>23</td>
<td>23</td>
<td>6</td>
<td>23</td>
<td>7</td>
<td>38</td>
<td>23</td>
<td>9</td>
</tr>
<tr>
<td>zhang</td>
<td>22</td>
<td>22</td>
<td>6</td>
<td>22</td>
<td>7</td>
<td>38</td>
<td>23</td>
<td>9</td>
</tr>
</tbody>
</table>
Appendix D

Histograms for Number of Rules Needed for Model 1–Model 8

with 10% Noise
Figure D1: Number of rules needed for Model 1 when noise is 10% for cosine, implication index, least contradiction, Kulczynski index, and PS

Note: min is the minimum, Q1 is the first quartile, median is the median, Q3 is the third quartile, and max is the maximum number of rules needed with 10% noise. Without noise is the number of rules needed when there is noise in the data, as shown in Table C2.
Figure D2: Number of rules needed for Model 2 when noise is 10%
for cosine, implication index, least contradiction, Kulczynski index, and PS

Note: min is the minimum, Q1 is the first quartile, median is the median, Q3 is the third quartile, and max is the maximum number of rules needed with 10% noise. Without noise is the number of rules needed when there is noise in the data, as shown in Table C2.
Figure D3: Number of rules needed for Model 3 when noise is 10% for cosine, implication index, least contradiction, Kulczynski index, and PS

Note: min is the minimum, Q1 is the first quartile, median is the median, Q3 is the third quartile, and max is the maximum number of rules needed with 10% noise. Without noise is the number of rules needed when there is noise in the data, as shown in Table C2.
Figure D4: Number of rules needed for Model 4 when noise is 10% for cosine, implication index, least contradiction, Kulczynski index, and PS

Note: min is the minimum, Q1 is the first quartile, median is the median, Q3 is the third quartile, and max is the maximum number of rules needed with 10% noise. Without noise is the number of rules needed when there is noise in the data, as shown in Table C2.
Figure D5: Number of rules needed for Model 5 when noise is 10% for cosine, implication index, least contradiction, Kulczynski index, and PS

Note: min is the minimum, Q1 is the first quartile, median is the median, Q3 is the third quartile, and max is the maximum number of rules needed with 10% noise. Without noise is the number of rules needed when there is noise in the data, as shown in Table C2.
Figure D6: Number of rules needed for Model 6 when noise is 10% for cosine, implication index, least contradiction, Kulczynski index, and PS

Note: min is the minimum, Q1 is the first quartile, median is the median, Q3 is the third quartile, and max is the maximum number of rules needed with 10% noise. Without noise is the number of rules needed when there is noise in the data, as shown in Table C2.
Figure D7: Number of rules needed for Model 7 when noise is 10% for cosine, implication index, least contradiction, Kulczynski index, and PS

Note: min is the minimum, Q1 is the first quartile, median is the median, Q3 is third quartile, and max is the maximum number of rules needed with 10% noise. Without noise is the number of rules needed when there is noise in the data, as shown in Table C2.

108
Figure D8: Number of rules needed for Model 8 when noise is 10% for cosine, implication index, least contradiction, Kulczynski index, and PS

Note: min is the minimum, Q1 is the first quartile, median is the median, Q3 is third quartile, and max is the maximum number of rules needed with 10% noise. Without noise is the number of rules needed when there is noise in the data, as shown in Table C2.
Appendix E

Matlab Codes
**E1: Matlab codes to find the number of rules needed for scheme 1**

```matlab
function record = addRank(record, tieType, p)
    for i=1:length(record)
        vec(i,1) = ceil(record(i).multi_cri(p)*1E+6);
        if isnan(vec(i,1))
            vec(i,1) = 0;
        end
        vec(i,2) = ceil(record(i).supp*1E+6);
    end
    [vec ind] = sortrows(vec,
         [-1 -2]);
    record(ind(1)).multi_rank(p) = 1;
    head = 1;
    record(ind(1)).num(p) = 1;
    if tieType == 0
        for i=2:length(record)
            if ~isinf(record(ic).multi_cri(p)) && ~isinf(record(ip).multi_cri(p))
                abs1 = abs(record(ic).multi_cri(p) - record(ip).multi_cri(p));
                elseif isnan(record(ic).multi_cri(p)) && isnan(record(ip).multi_cri(p))
                abs1 = 0.0;
                elseif ~isnan(record(ic).multi_cri(p)) && isnan(record(ip).multi_cri(p))
                abs1 = abs(record(ic).multi_cri(p));
                elseif isnan(record(ic).multi_cri(p)) && ~isnan(record(ip).multi_cri(p))
                abs1 = abs(record(ip).multi_cri(p));
            end
            abs2 = abs(record(ic).supp - record(ip).supp);
            if abs1 < 1E-10 && abs2 < 1E-10
                size(record(ic).rule,2) == size(record(ip).rule,2)
                record(ic).multi_rank(p) = record(ip).multi_rank(p);
            else
                for j=head:i
                    record(ind(j)).num(p) = i - head;
                end
                head = i;
                record(ic).multi_rank(p) = i;
            end
        end
        elseif tieType == 1
            j = 1;
            for i=2:length(record)
                record(ind(i)).num(p) = 1;
                record(ind(i)).multi_rank(p) = i;
                if ~isinf(record(ic).multi_cri(p)) && ~isinf(record(ip).multi_cri(p))
                    abs1 = abs(record(ic).multi_cri(p) - record(ip).multi_cri(p));
                elseif isnan(record(ic).multi_cri(p)) && isnan(record(ip).multi_cri(p))
                    abs1 = 0.0;
                elseif ~isnan(record(ic).multi_cri(p)) && isnan(record(ip).multi_cri(p))
                    abs1 = abs(record(ic).multi_cri(p));
                elseif isnan(record(ic).multi_cri(p)) && ~isnan(record(ip).multi_cri(p))
                    abs1 = abs(record(ip).multi_cri(p));
                end
                abs2 = abs(record(ic).supp - record(ip).supp);
                if abs1 < 1E-10 && abs2 < 1E-10
                    j = j + 1;
                else
                    for k=i-j+1
                        record(ind(k)).multi_rank(p) = sum(k) ;
                    end
                end
            end
        end
    end
end
```
record(ind(k)).num = i - head;
end
head = i;
end
j = 1;
end
else
for i=2:length(record)
    record(ind(i)).num = 1;
    record(ind(i)).multi_rank(p) = i;
end
end

File addRankAll
function record = addRankAll(record, tieType, mr)
for p=1:length(mr.multi_vec)
    record = addRank(record, tieType, p);
end
end

File bi2de
function d = bi2de(b, varargin)
% Typical error checking.
error(nargchk(1,3,nargin));
% --- Placeholder for the signature string.
sigStr = '\n';
flag = '\';
p = [];
% Check the type of the input B
if ~((isnumeric(b) || islogical(b)))
    error('The binary input must be numeric or logical.');
end
b = double(b);  % To allow logicals to work
% --- Identify string and numeric arguments
for i=1:length(varargin)
    if(i>1)
        sigStr(size(sigStr,2)+1) = '/';
    end
    % --- Assign the string and numeric flags
    if(ischar(varargin{i}))
        sigStr(size(sigStr,2)+1) = 's';
    elseif(isnumeric(varargin{i}))
        sigStr(size(sigStr,2)+1) = 'n';
    else
        error('Optional parameters must be string or numeric.');
    end
end
% --- Identify parameter signitures and assign values to variables
switch sigStr
% --- bi2de(d)
    case ''
% --- bi2de(d, p)
    case 'n'
        p = varargin{1};
% --- bi2de(d, flag)
    case 's'
        flag = varargin{1};
% --- bi2de(d, p, flag)
    case 'n/s'
        p = varargin{1};
        flag = varargin{2};
% --- bi2de(d, flag, p)
    case 's/n'
        flag = varargin{1};
        p = varargin{2};
% --- If the parameter list does not match one of these signatures.
    otherwise
        error('Syntax error.');
end
if isempty(b)
    error('Required parameter empty.');
end
if max(max(b < 0)) || max(max(~isfinite(b))) || (~isreal(b)) || ...
    (max(max(floor(b) ~= b))) || (~isreal(b)) || ...
    (max(max(floor(b) == b)))
    error('Input must contain only finite real positive integers.');
end
% Set up the base to convert from.
if isempty(p)
    p = 2;
else if max(size(p)) > 1
    error('Source base must be a scalar.');
elseif (floor(p) ~= p) || (~isfinite(p)) || (~isreal(p))
    error('Source base must be a finite real integer.);
elseif p < 2
    error('Source base must be greater than or equal to two.');
end
if max(max(b)) > (p-1)
    error('The elements of the matrix are larger than the base can represent.');
end
n = size(b,2);
% If a flag is specified to flip the input such that the MSB is to the left.
if isempty(flag)
    flag = 'right-msb';
elseif ~(strcmp(flag, 'right-msb') || strcmp(flag, 'left-msb'))
    error('Invalid string flag.');
end
if strcmp(flag, 'left-msb')
    b2 = b;
    b = b2(:,n:-1:1);
end
%%% The conversion
max_length = 1024;
pow2vector = p.^(0:1:(size(b,2)-1));
size_B = min(max_length,size(b,2));
d = b(:,1:size_B)*pow2vector(:,1:size_B).';
% handle the infs...
idx = find(max(b(:,max_length+1:size(b,2)).') == 1);
d(idx) = inf;

File candn2ov
function ret = candn2ov(candn, ncomb_vec)
    comb_size = length(ncomb_vec);
    divider = prod(ncomb_vec(2:end));
    num = candn;
    notyet = 1;
    for i=1:comb_size-1
        if num == 0
            ret(i) = ncomb_vec(i);
            if notyet
                ret(i-1) = ret(i-1) - 1;
                notyet = 0;
            end
        else
            ret(i) = fix(num/divider) + 1;
        end
        num = rem(num,divider);
        divider = divider/ncomb_vec(i+1);
    end
    if num == 0
        ret(comb_size) = ncomb_vec(comb_size);
        if notyet
            ret(comb_size-1) = ret(comb_size-1) - 1;
        end
    else
        ret(comb_size) = num;
    end
end

File computecri
function cri = computeCri(parm,choose)
    px = parm.px;
    py = parm.py;
    pab = parm.pab;
    panb = parm.panb;
    pnab = parm.pnab;
    sx = parm.sx;
    sab = parm.sab;
    cri = 0;
    switch choose
        case 1 %conf
            cri = pab/px;
        case 2 %lift
            cri = pab/(px*py);
    end
case 3 %PS
ceti = pab - px*py;
case 4 %speci
ceti = pnanb/(1-px);
case 5 %accur
ceti = pab + pnanb;
case 6 %lever
ceti = (pab/px) - (px*py);
case 7 %added
ceti = (pab/px) - py;
case 8 %relat
ceti = (pab/px)/(pnanb/(1-px));
case 9 %jacca
ceti = pab/(px-py-pab);
case 10 %acerta
ceti = (pab/px-py)/(1-py);
case 11 %yuleq
ceti = (pab*pnanb-pnanb*pnanb)/(pab*pnanb+pnanb*pnanb);
case 12 %yuley
ceti = sqrt(pab*pnanb-sqrt(pnanb*pnanb))/(sqrt(pab*pnanb)+sqrt(pnanb*pnanb));
case 13 %klog
ceti = sqrt(pab)/(pab/(px-1));
case 14 %apie
ceti = (pab+1)/(px+2);
case 15 %gini
ceti = px*(pab/(px-py))^2 + (pnanb/(1-px))^2 - (pnanb/(1-px))2 - (pab/px-1)^2;
case 16 %onesu
ceti = (pab/(px-py))log2(pab/(px-py));
case 17 %twosu
ceti = (pab/px-py)/(px-py);
case 18 %talg
ceti = (pab/px-py)/sqrt((1-py)^*(1-py));
case 19 %cosin
ceti = pab/sqrt(px-py);
case 20 %sinfor
ceti = log10(pab/(px-py));
case 21 %examp
ceti = (pab-pnanb)/py;
case 22 %cast
ceti = (pab-pnanb)/py;
case 23 %dimin
ceti = (pab-pnanb)*(1-py)(1-py)/((px-py)/(1-py));
case 24 %kappa
ceti = (pab*pnanb-pnanb-pnanb)/((1-py)(1-py));
case 25 %support
ceti = pab;
case 26 %eval
ceti = pab/px;
case 27 %convii
ceti = (pab+1)/(px-1);
case 28 %sebag
ceti = pab/px;
case 29 %oddmu
ceti = pab*(pab-px-py)/max(panb/panb);
case 30 %zhang
ceti = (pab*(1-py))/max(panb/panb);
case 31 %oddra
ceti = pab*(1-py)/max(panb/panb);
case 32 %jmeas
%cri = (pab*log10(pab/(px-py)) + (panb*log10(panb/(px-py))));
if pab == 0
ceti = cri + pab*log10(pab/(px-py));
end
if pnanb == 0
ceti = cri + pnanb*log10(pnanb/(px-py));
end
end
case 33 %twova
%cri = (pab*log2(pab/(px-py)) + (pnanb*log2(pnanb/(1-py))));
if pab == 0
ceti = cri + pab*log2(pab/(px-py));
end
if pnanb == 0
ceti = cri + pnanb*log2(pnanb/(1-py));
end
end
case 34 %twove
cri = cri + pnanb*log2(pnanb/((1-px)*(1-py)));
end

case 34 %ganas
cri = (pab-panb)/px;
end
case 35 %impli
cri = -(panb-(px*(1-py)))/sqrt(px*(1-py));
case 36 %direc
cri = (pab-(px*py))/sqrt(px*py);
case 37 %aroger
cri = (1-panb-pnb)/(1+panb+panb);
case 38 %dice
cri = pab/(pab+0.5*(pnb+panb));
case 39 %kulcz
cri = 0.5*(pab/pxy)+(pab/pxy);
case 40 %norma
%cri = ((pab*log2(pab/((1-px)*py)))+(panb*log2(panb/((1-px)*py)))+(pnab*log2(pnab/((1-px)*py)))+(pnanb*log2(pnanb/((1-px)*(1-py))));
if pab ~= 0
cri = cri + (pab*log2(pab/(px*py)))/-((px*log2(px))+((1-px)*log2(1-px)));
end
if panb ~= 0
cri = cri + (panb*log2(panb/(px*(1-py))))/-(px*log2(px))+((1-px)*log2(1-px));
end
if pnab ~= 0
cri = cri + (pnab*log2(pnab/((1-px)*py)))/-((px*log2(px))+((1-px)*log2(1-px)));
end
if pnanb ~= 0
cri = cri + (pnanb*log2(pnanb/((1-px)*(1-py))))/-(px*log2(px))+((1-px)*log2(1-px));
end
otherwise %conf
cri = pab/px;
end
end

% Function de2bi
function b = de2bi(varargin)
% Typical error checking.
error(nargchk(1,4,nargin));
% --- Placeholder for the signature string.
sigStr = '';
flag = 's';
p = [];
n = [];
% --- Identify string and numeric arguments
for i=1:nargin
if(i>1)
sigStr(size(sigStr,2)+1) = '/';
end;
% --- Assign the string and numeric flags
if(ischar(varargin{i}))
sigStr(size(sigStr,2)+1) = 's';
else if(isnumeric(varargin{i}))
sigStr(size(sigStr,2)+1) = 'n';
else
error('Only string and numeric arguments are accepted.');
end;
end;
% --- Identify parameter signitures and assign values to variables
switch sigStr
% --- de2bi(d)
% case 'n'
d = varargin{1};
% --- de2bi(d, n)
case 'n/n'
d = varargin{1};
n = varargin{2};
% --- de2bi(d, flag)
case 'n/s'
d = varargin{1};
flag = varargin{2};
% --- de2bi(d, n, flag)
case 'n/n/s'
d = varargin{1};
flag = varargin{2};
n = varargin{3};
% --- de2bi(d, flag, n)
case 'n/n/s'
d = varargin{1};
n = varargin{2};
flag = varargin{3};
% --- de2bi(d, flag, n)
case 'n/n/s'
d = varargin{1};
n = varargin{2};
flag = varargin{2};
n = varargin{3};

% --- de2bi(d, n, p)
case 'n/n/n'
d = varargin{1};
n = varargin{2};
p = varargin{3};
% --- de2bi(d, n, p, flag)
case 'n/n/n/s'
d = varargin{1};
n = varargin{2};
p = varargin{3};
flag = varargin{4};
% --- de2bi(d, n, flag, p)
case 'n/n/s/n'
d = varargin{1};
flag = varargin{2};
n = varargin{3};
p = varargin{4};
% --- de2bi(d, flag, n, p)
case 'n/s/n/n'
d = varargin{1};
flag = varargin{2};
n = varargin{3};
p = varargin{4};

% --- If the parameter list does not match one of these signatures.
otherwise
    error('Syntax error.');
end;
if isempty(d)
    error('Required parameter empty.');
end

if isempty(d)
    len_d = length(d);
    if max(max(d < 0)) | max(max(~isfinite(d))) | (~isreal(d)) | (max(max(floor(d) ~= d))
        error('Input must contain only finite real positive integers.');
    end

    % Assign the base to convert to.
    if isempty(p)
        p = 2;
    elseif max(size(p) ~= 1)
        error('Destination base must be scalar.');
    elseif (~isfinite(p)) | (~isreal(p)) | (floor(p) ~= p)
        error('Destination base must be a finite real integer.');
    elseif p < 2
        error('Cannot convert to a base of less than two.');
    end;
    % Determine minimum length required.
    tmp = max(d);
    if tmp == 0
        ntmp = floor( log(tmp) / log(p) ) + 1;
    else
        ntmp = 1;
    end
    % This takes care of any round off error that occurs for really big inputs.
    if (~ (p^ntmp) > tmp )
        ntmp = ntmp + 1;
    end
    % Assign number of columns in output matrix.
    if isempty(n)
        n = ntmp;
    elseif max(size(n) ~= 1)
        error('Specified number of columns must be scalar.');
    elseif (~isfinite(n)) | (~isreal(n)) | (floor(n) ~= n)
        error('Specified number of columns must be a finite real integer.');
    elseif n < ntmp
        error('Specified number of columns in output matrix is too small.');
    end
    % Check if the string flag is valid.
    if isempty(flag)
        flag = 'right-msb';
    elseif ~(strcmp(flag, 'right-msb') | strcmp(flag, 'left-msb'))
        error('Invalid string flag.');
    end
    % Initial value.
b = zeros(len_d, n);
    % Perform conversion.
    for i = 1 : len_d
        % Cycle through each element of the input vector/matrix.
\[ j = 1; \]
\[ \text{tmp = d(i);} \]
\[ \text{while (j <= n) \& (tmp > 0) % Cycle through each digit.} \]
\[ \text{b(i,j) = \text{rem}(\text{tmp}, p); % Determine current digit.} \]
\[ \text{tmp = \text{floor}(\text{tmp}/p);} \]
\[ j = j + 1; \]
\[ \text{end;} \]
\[ \text{end;} \]
\[ \% If a flag is specified to flip the output such that the MSB is to the left. \]
\[ \text{if strcmp(flag, 'left-msb')} \]
\[ \text{b2 = b;} \]
\[ \text{b = b2(:,n:-1:1);} \]
\[ \text{end} \]
\[ \% [EOF] de2bi.m \]

File `filterParents`

```matlab
function rrcell = filterParents(rcell)
    rrcell = rcell;
    [nrcell,level] = size(rcell);
    if nrcell < 2
        return;
    end
    for s=2:nrcell
        for y=1:level
            rec = rcell{s,y};
            sp1 = floor(s/2);
            sp2 = s - sp1;
            recp1 = rcell{sp1,y};
            recp2 = rcell{sp2,y};
            rrcell{s,y} = subFilterParents(rec,recp1,recp2);
        end
    end
end
```

File `filterWithY`

```matlab
function [yxa,yxr] = filterWithY(y, xdata, record, yx, x)
    yxa = [ ]; yxr = [ ];
    n = length(yx);
    if n > 0
        for i=1:n
            k = yx(i);
            rule = record(k).rule;
            rule(:,end) = [ ];
            yn = maskY(xdata,rule,x);
            testv = y - yn;
            if x == 0
                res = find(testv > 0);
            else
                res = find(testv < 0);
            end
            if isempty(res)
                yxa = [yxa k];
            end
        end
        yxr = setdiff(yx,yxa);
    end
end
```

File `findRealK`

```matlab
function realK = findRealK(record,topK)
    realK = 0;
    if record(topK).multi_rank == topK
        realK = topK;
    else
        for i=topK:length(record)
            if topK < record(i).multi_rank
                realK = i-1;
                return
            end
        end
    end
end
```

File `fixRank`

```matlab
function rec = fixRank(record)
    for i=1:length(record)
        record(i).rank_cri = record(i).multi_rank(1);
    end
vec = [record.rank_cri];
```
function ret = genComb(i, comb)
    comblen = size(comb,2);
    offset = 2^comblen;
    r = rem(i-1,offset);
    q = ceil(i/offset);
    up = comb(q,:);
    down = de2bi(r,comblen, 'left-msb');
    ret = [up;down];
end

function rcell = genRules(data, minsupp, rmax, level, chc, mr)
    to1 = 1 - level(1);
    for y=level(1):level(2)
        for r=1:rmax
            rcell{r,y+to1} = subGenRules(data, minsupp, r, y, chc, mr);
        end
    end
end

function ret = genTerm(xdata, rule_vec)
    nvar = size(xdata, 2);
    comb_size = length(rule_vec);
    ncand = 1;
    for i=1:comb_size
        rcand(i).rule_len = rule_vec(i);
        rcand(i).comb = nchoosek(1:nvar, rule_vec(i));
        rcand(i).ncomb = length(rcand(i).comb)*2^rule_vec(i);
        ncand = ncand*rcand(i).ncomb;
    end
    n = 0;
    ret = struct('term',{ });
    for i=1:ncand
        cvec = [rcand.ncomb];
        ivec = candntov(i, cvec);
        skip = 0;
        for j=1:length(cvec)-1
            if cvec(j) == cvec(j+1) && ivec(j) >= ivec(j+1)
                skip = 1;
                continue;
            end
        end
        if skip
            continue;
        end
        term = genComb(ivec(1),rcand(1).comb);
        for j=2:length(ivec)
            rule = genComb(ivec(j),rcand(j).comb);
            if isParent2(term,rule)
                skip = 1;
                continue;
            end
            term = [term [0;0] rule];
        end
        if skip
            continue;
        end
        n = n + 1;
        ret(n).term = term;
    end
end

function ret = genY(xdata, term)
data size = size(xdata, 1);
ind = find(term(1,:) == 0);
nrule = length(ind) + 1;
ind = [0 ind size(term,2)+1];
for i=1:datasize
    y(i) = 0;
    for j=1:nrule
        range = ind(j)+1:ind(j+1)-1;
        rule = term(:,range);
        if xdata(i,rule(1,:)) == rule(2,:)
            y(i) = 1;
        end
    end
end
ret = y';

File getIndRec

function indRec = getIndRec(record, cterm)
    nterm = length(cterm);
    for i=1:nterm
        indRec(i) = 0;
    end
    count = 0;
    for i=1:length(record)
        recsize = size(record(i).rule,2);
        rule = record(i).rule;
        rule(:,end) = [];
        for j=1:nterm
            if size(cterm{j},2) == recsize
                cterm = cterm{j};
                cterm(:,end) = [];
                if cterm == rule
                    indRec(j) = i;
                    count = count+1;
                end
            end
        end
    end
    if count == length(cterm)
        return;
    end
end

File getLogitTable

function [ret, tm] = getLogitTable(data, record, comb, i, K, topK, out)
    tm = 0;
    y = data(:,end);
    nvar = size(data,2)-1;
    realK = min(topK,length(record));
    for l=1:length(record)
        rec = record(l);
        if topK < rec.rank_cri;
            realK = l-1;
            break;
        end
    end
    topK = realK;
    str = sprintf('i = %d, topK = %d', i, topK);
    disp(str);
    lmax = -inf;
    ind = nchoosek(1:topK,K);
    for j=1:size(ind,1),
        r = ind(:,j);
        vars = {record(r).rule};
        x = getx(data, vars);
        l = logit(x,y);
        if l > lmax
            lmax = l;
            vmax = vars;
            vmax = sortc(vmax,nvar);
            rank = r;
        end
    end
    tm = i;
    ret.K = i;
    ret.comb = comb;
    ret.rank = rank;
    ret.logit = lmax;
ret.vars = vmax;
ret.svars = textVars2(vmax,1);
end

File getProp

function [px, py] = getProp(data, term)
  datasize = size(data,1);
  py = sum(data(:,end))/datasize;
  ind = find(term(:,1)==0);
  nrule = length(ind) + 1;
  ind = [0 ind size(term,2)+1];
  ret = {1};
  for i=1:nrule
    range = ind(i)+1:ind(i+1)-1;
    rule = term(:,range);
    subdata = data(:, rule(1,:));
    count = 0;
    for j=1:datasize
      if subdata(j,:) == rule(2,:)
        count = count + 1;
      end
    end
    px(i) = count/datasize;
  end
end

File getRule

%%% function to extract the rules from the input terms
%%% term - input term
function ret = getRule(term)
  text = textTerm(term);
  loc = find(text == ',');
  loc = [0 loc length(text)+1];
  nrule = length(loc)-1;
  ret = cell(1,nrule);
  for i=1:nrule
    ret{i} = text(loc(i)+1:loc(i+1)-1);
  end
end

File getx

function ret = getx(data, vars)
  k = size(vars,2);
  datasize = size(data,1);
  for j=1:k,
    var = vars{j}(:,1:end-1);
    up = var(1,:);
    down = var(2,:);
    for i=1:datasize,
      if data(i,up) == down
        ret(i,j) = 1;
      else
        ret(i,j) = 0;
      end
    end
  end
end

File getY

function yn = getY(k,x,record,xdata)
  yn = [ ];
  rule = record(k).rule;
  rule(:,end) = [ ];
  yn = maskY(xdata,rule,x);
end

File isAccept

function ii = isAccept(y,yn,x)
  testv = y - yn;
  if x == 0
    res = find(testv > 0);
  else
    res = find(testv < 0);
  end
  ii = isempty(res);
end

120
function [flag, h] = isDuplicate(px, py, prop)
    psize = size(prop,1);
    flag = 0;
    h = 0;
    if isempty(prop)
        return;
    end
    for i=1:psize
        if py == prop(i,end)
            pp = prop(i,1:end-1);
            tmp1 = intersect(px,pp);
            tmp2 = intersect(px,pp,'rows');
            if length(tmp1) > length(tmp2)
                tmp = tmp1;
            else
                tmp = tmp2;
            end
            if length(px) == length(tmp)
                if px == tmp
                    flag = 1;
                    h = i;
                    return;
                end
            end
        end
    end
end

function yxy = isMatchY(y, yxm)
    test = xor(y,yxm);
    if sum(test) == 0
        yxy = 1;
    else
        yxy = 0;
    end
end

function ret = isParent(rule_child, rule_parent, ended)
    if ended
        rule_child(:,end) = [];
        rule_parent(:,end) = [];
    end
    size_child = size(rule_child,2);
    size_parent = size(rule_parent,2);
    if size_child == 1 && size_parent == 1 && rule_child(1,1) <= rule_parent(1,1)
        ret = true;
        return;
    end
    if sum(intersect(rule_child(1,:), rule_parent(1,:))) == sum(rule_parent(1,:))
        ind = nchoosek(1:size_child,size_parent);
        for i=1:size(ind,1)
            if rule_child(:,ind(i,:)) == rule_parent
                ret = true;
                return;
            end
        end
    end
    ret = false;
end

function ret = isParent2(rule_parent_vec, rule_child)
    loc = find(rule_parent_vec(1,:) == 0);
    size_vec = size(rule_parent_vec,2);
    loc = [0 loc size_vec+1];
    if isempty(loc)
        ret = isParent(rule_child,rule_parent_vec,0);
        return;
    end
    for i=1:length(loc)-1
        rind = loc(i+1)-loc(i+1)-1;
        rule_parent = rule_parent_vec(:,rind);
        if isParent(rule_child,rule_parent)
            ret = true;
        end
    end
end
function ret = logit(x,y)
%set of explanatory variables
%x=[data(:,1),data(:,i),data(:,j)];
%An initial values for coefficient estimates
[lvars, nvars] = size(x);
ret = false;
end

end

File logit

File mainScript

clear;
format short;
infile = 'data/data.data';  % change input filename here
out.outfile = 'results/rank3-3-40.csv';
%outlogit = 'results/logit.csv';
outmulti = 'results/multirank3-3-40.csv';
minsupp = 0.005;            % change minimum support here
out.checkterm = [1:10000];
tieType = 0;                % 0 for min rank, 1 for average, 2 for none
xdata = load(infile);
level = [0 1];              % possible response
rule_vec = [3 3];           % size of combination wanted to find for log likelihood
maxK = 1000;                % maximum K to output
topK = 30;                  % top K rules
chc = 40;                   % chosen criteria to rank topK
maxn = 4;                   % maximum #of rule size
mr.multi_enable = 1;        % enable multirank output
mr.multi_vec = [1 2];       % output ranking of other criteria
% chc is included as a default
outmultiHeaderScript();
tic;
rec_term = genTerm(xdata, rule_vec);
cnt = 0; prop = [];
out.verbose = 1;
report = zeros(length(out.checkterm),4);
report(:,2) = topK;
recount = 1;
k0 = []; k1 = [];
r0 = []; r1 = [];
for i=1:length(rec_term)
term = rec_term(i).term;
y = genY(xdata, term);
% append y to data

end

end
```matlab
data = [xdata y];
% begin check propability %
[px, py] = getProp(data, term);
x = length(px);
for j=1:nx
    allProp(i,j) = px(j);
end
allProp(i,nx+1) = py;
% discard if all y's are 0s or 1s
if py == 0 || py == 1
    continue;
end
% discard if propability is the same as someone before
[flag, h] = isDuplicate(px, py, prop);
if flag
    c(h) = c(h)+1;
    history(h,c(h)) = i;
    continue;
end
cnt = cnt + 1;
history(cnt,1) = i;
c(cnt) = 1;
for j=1:nx
    prop(cnt,j) = px(j);
end
prop(cnt,nx+1) = py;
% end check propability %
rcell = genRules(data, minsupp, rmax, level, chc, mr);
% rcell = filterParents(rcell);
record = mergeRcell(rcell, level, size(data, 2));
record = sortCri(record, tieType);
if mr.multi_enable
    record = addRankAll(record, tieType, mr);
end
% record = fixRank(record);
rule_cell = getRule(term);
comb = '';
for s=1:length(rule_cell)
    comb = [comb ' ' rule_cell{s}];
end
comb(1) = ' ';
%[table(cnt), tm] = getLogitTable(data, record, comb, i, K, topK, out);
outrankScript();
oumtiScript();
elapsed_time = toc;
if mr.multi_enable
    fclose(fp);
end
printReportScript();
disp('Simulation Completed');
% outlogitScript();
```

File **maskY**

```matlab
function yn = maskY(xdata, rule, x)
yn = []; datasize = size(xdata, 1);
for i=1:datasize
    if rule(2,:) == xdata(i, rule(1,:))
        if x == 0
            yn = [yn; 0];
        else
            yn = [yn; 1];
        end
    else
        if x == 0
            yn = [yn; 1];
        else
            yn = [yn; 0];
        end
    end
end
```

File **mergeRcell**

```matlab
function record = mergeRcell(rcell, level, yn)
record = [ ];
[r,c] = size(rcell);
```
for i=1:r
    for j=1:c
        l = level(1):level(2);
        rec = rcell{i,j};
        for k=1:length(rec)
            rec(k).rule = [rec(k).rule [yn;l(j)]];
            record = [record rec];
        end
    end
end

File mergeY

function yxm = mergeY(y, xdata, record, yxa, x, p, out)
    n = length(yxa);
    datasize = size(xdata,1);
    ync = {};
    if x == 0
        yxm = ones(datasize,1);
    else
        yxm = zeros(datasize,1);
    end
    if n > 0
        for i=1:n
            k = yxa(i);
            rule = record(k).rule;
            rule(:,end) = [];
            yn = maskY(xdata,rule,x);
            ync = [ync yn];
            if x == 0
                yxm = yxm & yn;
            else
                yxm = yxm | yn;
            end
        end
    end
    if out.verbose > 1
        if x == 0
            str = ['results/y' num2str(p) '-0.csv'];
        else
            str = ['results/y' num2str(p) '-1.csv'];
        end
        dlmwrite(str, [y yxm ync], 'delimiter', ',');
    end
end

File outlogitScript

fp = fopen(outlogit, 'wt');
%header%
fprintf(fp, 'combination, condition, select, correct, logit');
for i=1:length(table)
    t = table(i);
    fprintf(fp, '
%d, %s, %s, %d, %f', t.K, strrep(t.comb,' ','|'), ...
        strrep(t.svars,' ','|'), strcmp(t.comb,t.svars), t.logit);
end
fclose(fp);

File outmultiHeaderScript

if mr.multi_enable
    rh{1} = 'conf';   rh{2} = 'lift';   rh{3} = 'PS';
    rh{4} = 'speci';  rh{5} = 'accur';  rh{6} = 'lever';
    rh{7} = 'added';  rh{8} = 'relat';  rh{9} = 'aceca';
    rh{10} = 'certa'; rh{11} = 'yuleq'; rh{12} = 'yuley';
    rh{13} = 'klosy'; rh{14} = 'lipla'; rh{15} = 'gini';
    rh{16} = 'onesu'; rh{17} = 'twous'; rh{18} = 'taub';
    rh{19} = 'cosin'; rh{20} = 'infor'; rh{21} = 'least';
    rh{22} = 'exam';  rh{23} = 'coole'; rh{24} = 'kappa';
    rh{25} = 'support'; rh{26} = 'recol'; rh{27} = 'convi';
    rh{28} = 'sebag'; rh{29} = 'oddru'; rh{30} = 'zhang';
    rh{31} = 'impni'; rh{32} = 'jmea'; rh{33} = 'twova';
    rh{34} = 'ganas'; rh{35} = 'impli'; rh{36} = 'direc';
    rh{37} = 'roger'; rh{38} = 'dice'; rh{39} = 'kulcz';
    rh{40} = 'norma';
    mr.multi_vec = [chc setdiff(mr.multi_vec,chc)];
    fp = fopen(outmulti, 'wt');
    fprintf(fp, '%term#,%term');
    fprintf(fp, ',num, accept, realK');
    fprintf(fp, '[chc setdiff(mr.multi_vec,chc)];
    fp = fopen(outmulti, 'wt');
    fprintf(fp, '%term#,%term');
    fprintf(fp, ',num, accept, realK');
    n = length(mr.multi_vec);
if n > 1
    fprintf(fp, ',
    for i=2:n
        mv = mr.multi_vec(i);
        fprintf(fp, ', %s', rh{mv});
    end
    fprintf(fp, ',
end
fprintf(fp, ', lrank, num_lrank
fprintf(fp, '
end
File outmultiScript

if mr.multi_enable
    if y0y & y1y
tres = 2;
else if y0y & ~y1y
tres = 0;
elseif ~y0y & y1y
tres = 1;
else
tres = 3;
end
str = sprintf('i = %d', i);
disp(str);
n = length(mr.multi_vec);
dd = find(term(1,:)==0);
nterm = length(dd)+1;
dd = [0 dd size(term,2)+1];
for ii=1:nterm
cterm(ii) = [term(:,dd(ii)+1:dd(ii+1)-1) [0;0]];
end
indRec = getIndRec(record, cterm);
for ii=1:length(indRec)
    if ii == 1
        fprintf(fp, '%d', i);
    else
        fprintf(fp, '
    end
    fprintf(fp, ', %s', textVars2(cterm(ii),1));
    if indRec(ii) == 0
        multi_rank = zeros(1,n);
    else
        multi_rank = record(indRec(ii)).multi_rank;
        num = record(indRec(ii)).num(1);
    end
    fprintf(fp, ', %f', multi_rank(1));
    fprintf(fp, ', %d', num);
    if ii == 1 & isempty(setdiff(i,out.checkterm))
        fprintf(fp, ', %d, %d', tres, report(rcount-1,3));
    end
    if n > 1
        fprintf(fp, ',
        if ii > 1
            fprintf(fp, ',
        end
        for jj=2:n
            fprintf(fp, ', %f', multi_rank(jj));
        end
        fprintf(fp, ',
    end
    fprintf(fp, ',
    if ii == 1
        fprintf(fp, ', %f, %f, k0(rcount-1), n0(rcount-1));
    else
        fprintf(fp, ', %f, %f, k1(rcount-1), n1(rcount-1));
    end
    fprintf(fp, ',
end
fprintf(fp, '
end
File outrankScript

ict = find(out.checkterm==i);
if ~isempty(ict)
    realK = findRealK(record,topK);
    report(rcount,1) = i;
    report(rcount,3) = realK;
y0 = []; y1 = [];
for j=1:length(record)
    if record(j).rule(2,end) == 0
        y0 = [y0 j];
    else
        y1 = [y1 j];
    end
end
%for y -> 0
k = 0;
yu = ones(length(y),1);
for j=1:length(y0)
    yn = getY(y0(j),0,record,xdata);
    if isAccept(y,yn,0)
        yu = yUnion(yn,yu,0);
        if y == yu
            k = y0(j);
            break;
        end
    end
end
k0 = [k0 record(k).multi_rank(1)];
n0 = [n0 record(k).num(1)];
%for y -> 1
k = 0;
yu = zeros(length(y),1);
for j=1:length(y1)
    yn = getY(y1(j),1,record,xdata);
    if isAccept(y,yn,1)
        yu = yUnion(yn,yu,1);
        if y == yu
            k = y1(j);
            break;
        end
    end
end
k1 = [k1 record(k).multi_rank(1)];
n1 = [n1 record(k).num(1)];
% old one
y0 = []; y1 = [];
for j=1:realK
    if record(j).rule(2,end) == 0
        y0 = [y0 j];
    else
        y1 = [y1 j];
    end
end
[y0a,y0r] = filterWithY(y,xdata,record,y0,0);
[y1a,y1r] = filterWithY(y,xdata,record,y1,1);
yr = union(y0r,y1r);
report(rcount,4) = length(y0a) + length(y1a);
report(rcount,5) = length(y0);
report(rcount,6) = length(y0a);
report(rcount,7) = length(y1);
report(rcount,8) = length(y1a);
rcount = rcount + 1;
%Uncomment this line to see all the indexes of rejected rules
%disp(yr);
if out.verbose > 0
    %WRITE TO FILE
    idot = strfind(out.outfile,'.);
    outf = [];
    outf = [out.outfile(1:idot-1) num2str(out.checkterm(ict)) out.outfile(idot:end)];
    fprintf(fout, 'rank, rule, response, criteria, support, num, status');
    nrank = length(record(1).multi_rank);
    fprintf(fout, ',
    for k=2:nrank
        fprintf(fout, ',', rank, val);
    end
    if maxK > length(record)
        maxK = length(record);
    end
    for j=1:maxK
        rec = record(j);
        varstr = textVars2({rec.rule},0);
        idy = strfind(varstr,'Y');

fprintf(fpor, 'm%f, %s, %s, %f, %f, %d', rec.multi_rank(1),...
  varstr(1:idy-1), varstr(idy:end), rec.cri, rec.supp, rec.num(1));
if j<=realK
  r = j;
  if isempty(yr)
    r = setdiff(j,yr);
  end
  if isempty(r)
    fprintf(fpor, ', r');
  else
    fprintf(fpor, ', a');
  end
end
if nrank > 1
  fprintf(fpor, ',');
  if j > realK
    fprintf(fpor, ',');
  end
  for k=2:nrank
    fprintf(fpor, ', %f, %f', rec.multi_rank(k), rec.multi_cri(k));
  end
end
close(fpor);
%
merge the class
y0m = mergeY(y, xdata, record, y0a, 0, i, out);
y1m = mergeY(y, xdata, record, y1a, 1, i, out);
y0y = isMatchY(y, y0m);
y1y = isMatchY(y, y1m);
%
File printReportScript

outrep = 'results/multireport.csv';
fprep = fopen(outrep, 'wt');
fprintf(fprep, 'term, topK, realK, left, drop, y0, y0a, y1, y1a\n');
for i=1:rcount-1;
  fprintf(fprep, '%d, %d, %d, %d, %d, %d, %d, %d, %d\n', ...
    report(i,1), report(i,2), report(i,3), report(i,4), report(i,3) - report(i,4), ...
    report(i,5), report(i,6), report(i,7), report(i,8));
end
close(fprep);
%
File sortCri

% change to new name if you want to make another sort i.e. sortPS()
function ret = sortCri(record, tieType)
  vec = [record.supp];
  [~, ind] = sort(vec,'descend');
  rec = record(ind);
  vec = [rec.cri];
  [~, ind] = sort(vec,'descend');
  rec = rec(ind);
  rec(1).rank_cri = 1;
  if tieType == 0
    for i=2:length(rec)
      abs1 = abs(rec(i).cri - rec(i-1).cri);
%abs2 = abs(rec(i).supp - rec(i-1).supp);
      %if abs1 < 1E-10 && abs2 < 1E-10
      if  abs1 < 1E-10
        rec(i).rank_cri = rec(i-1).rank_cri;
      else
        rec(i).rank_cri = i;
      end
    end
  end
elseif tieType == 1
    j = 1;
    for i=2:length(rec)
        if i == 1
            rec(i).rank_cri = i;
        else
            abs1 = abs(rec(i).cri - rec(i-1).cri);
            if abs1 < 1E-10
                j = j + 1;
            else
                if j > 1
                    sum = rec(i-1).rank_cri + rec(i-j).rank_cri;
                    for k=i-j+1:i-1
                        rec(k).rank_cri = sum/2;
                    end
                    j = 1;
                end
            end
        end
    end
    ret = rec;
end

File subFilterParents

function srec = subFilterParents(rec,recp1,recp2)
    eind = [];
    s = size(rec(1).rule,2);
    sp1 = size(recp1(1).rule,2);
    sp2 = size(recp2(1).rule,2);
    ind = nchoosek(1:s,sp1);
    for i=1:length(rec)
        skip = 0;
        rule = rec(i).rule;
        cri = rec(i).cri;
        supp = rec(i).supp;
        rulep1 = rule(:,ind(j,:));
        rulep2 = rule;
        rulep2(:,ind(j,:)) = [];
        for k=1:length(recp1)
            if rulep1 == recp1(k).rule
                if cri < recp1(k).cri
                    skip = 1;
                    eind = [eind i];
                    continue;
                elseif cri == recp1(k).cri && supp <= recp1(k).supp
                    skip = 1;
                    eind = [eind i];
                    continue;
                end
            end
        end
        for k=1:length(recp2)
            if rulep2 == recp2(k).rule
                if cri < recp2(k).cri
                    skip = 1;
                    eind = [eind i];
                    continue;
                elseif cri == recp2(k).cri && supp <= recp2(k).supp
                    skip = 1;
                    eind = [eind i];
                    continue;
                end
            end
        end
        if skip
            continue;
        end
    end
    srec = rec;
    srec(eind) = [];
end
function rec = subGenRules(data, minsupp, r, y, chc, mr)
    [datasize, nvar] = size(data);
    nvar = nvar - 1;
    parm.py = sum(data(:, end)==y)/datasize;
    ind = nchoosek(1:nvar,r);
    cnt = 0;
    for i=1:size(ind,1)
        n = 2^r;
        for j=0:n-1
            rule = [ind(i,:);dec2bin(j,r)];
            sx = 0;
            sab = 0; sanb = 0;
            snab = 0; snanb = 0;
            for k=1:datasize
                if data(k,rule(1,:)) == rule(2,:)
                    sx = sx + 1;
                    if data(k,end) == y
                        sab = sab + 1;
                    else
                        sanb = sanb + 1;
                    end
                else
                    if data(k,end) == y
                        snab = snab + 1;
                    else
                        snanb = snanb + 1;
                    end
                end
            end
            parm.px = sx/datasize;
            parm.pab = sab/datasize;
            parm.panb = sanb/datasize;
            parm.pnab = snab/datasize;
            parm.sx = sx;
            parm.sab = sab;
            if parm.pab >= minsupp
                cnt = cnt + 1;
                rec(cnt).rule = rule;
                rec(cnt).supp = parm.pab;
                rec(cnt).cri = computeCri(parm,chc);
                rec(cnt).rank_cri = 0;
                if mr.multi_enable
                    multi_cri = [];
                    for q=1:length(mr.multi_vec)
                        tmp = computeCri(parm,mr.multi_vec(q));
                        multi_cri = [multi_cri tmp];
                    end
                    rec(cnt).multi_cri = multi_cri;
                    rec(cnt).multi_rank = zeros(1,length(mr.multi_vec));
                end
            end
        end
    end
    end
    parm.px = sx/datasize;
    parm.pab = sab/datasize;
    parm.panb = sanb/datasize;
    parm.pnab = snab/datasize;
    parm.sx = sx;
    parm.sab = sab;
    if parm.pab >= minsupp
        cnt = cnt + 1;
        rec(cnt).rule = rule;
        rec(cnt).supp = parm.pab;
        rec(cnt).cri = computeCri(parm,chc);
        rec(cnt).rank_cri = 0;
        if mr.multi_enable
            multi_cri = [];
            for q=1:length(mr.multi_vec)
                tmp = computeCri(parm,mr.multi_vec(q));
                multi_cri = [multi_cri tmp];
            end
            rec(cnt).multi_cri = multi_cri;
            rec(cnt).multi_rank = zeros(1,length(mr.multi_vec));
        end
    end
end
end

File textTerm

function ret = textTerm(term)
    n = size(term,2);
    ind = find(term(1,:) == 0);
    nterm = length(ind) + 1;
    ind = [0 ind n+1];
    ret = [];
    for i=1:nterm
        range = ind(i)+1:ind(i+1)-1;
        rule = term(:,range);
        for j=1:size(rule,2)
            ret = [ret 'X' int2str(rule(1,j)) '(' int2str(rule(2,j)) ')'];
        end
        if i ~= nterm
            ret = [ret ','];
        end
    end
end
end

File textVars2

%%% this function is called by outputScript
function svars = textVars2(vars, nY)
    len = size(vars,2);
    svars = [];
    for i=1:len,
        fin = '\';
        var = vars{i};
        sizevar = size(var,2)-1;
        for j=1:sizevar
            temp = sprintf('X%d(%d)', var(1,j), var(2,j));
            fin = [fin temp];
        end
        temp = sprintf('Y(%d)', var(2,end));
        fin = [fin temp];
        svars = [svars ' ' fin];
    end
    svars(1) = [];
    if nY
        svars = svars(:,1:end-4);
        ind = strfind(svars,' ');
        for i=length(ind):-1:1
            svars = [svars(:,1:ind(i)-5) svars(:,ind(i):end)];
        end
    end
end

File yUnion
function yu = yUnion(y,yn,x)
    if x == 0
        yu = bitand(y,yn);
    else
        yu = bitor(y,yn);
    end
end

E2: Matlab codes to find the number of rules needed for scheme 2

The difference from scheme 1 is file sortCri which is shown below.

File sortCri

function ret = sortCri(record, tieType)
    vec = [record.supp];
    [~, ind] = sort(vec,'descend');
    rec = record(ind);
    indNaN = find(isnan(vec));
    vec(indNaN) = 0;
    [~, ind] = sort(vec,'descend');
    rec = rec(ind);
    rec(1).rank_cri = 1;
    if tieType == 0
        for i=2:length(rec)
            if ~isnan(rec(i).cri) && ~isnan(rec(i-1).cri)
                abs1 = abs(rec(i).cri - record(i-1).cri);
                elseif isnan(rec(i).cri) && isnan(rec(i-1).cri)
                    abs1 = 0.0;
                elseif ~isnan(rec(i).cri) && ~isnan(rec(i-1).cri)
                    abs1 = abs(record(i).cri);
                else
                    abs1 = abs(record(i-1).cri);
                end
                abs2 = abs(rec(i).supp - rec(i-1).supp);
                if abs1 < 1E-10 && abs2 < 1E-10
                    rec(i).rank_cri = rec(i-1).rank_cri;
                else
                    rec(i).rank_cri = i;
                end
            end
        end
    elseif tieType == 1
        j = 1;
        for i=2:length(rec)
rec(i).rank_cri = i;
if ~isinf(rec(i).cri) && ~isinf(rec(i-1).cri)
    if ~isnan(rec(i).cri) && ~isnan(rec(i-1).cri)
        abs1 = abs(rec(i).cri - record(i-1).cri);
    elseif isnan(rec(i).cri) && isnan(rec(i-1).cri)
        abs1 = 0.0;
    elseif ~isnan(rec(i).cri) && isnan(rec(i-1).cri)
        abs1 = abs(record(i).cri);
    elseif isnan(rec(i).cri) && ~isnan(rec(i-1).cri)
        abs1 = abs(record(i-1).cri);
    end
    abs2 = abs(rec(i).supp - rec(i-1).supp);
    if abs1 < 1E-10 && abs2 < 1E-10
        j = j + 1;
    elseif abs1 > 1
        sum = rec(i-1).rank_cri + rec(i-j).rank_cri;
        for k=i-j+1 to i
            rec(k).rank_cri = sum/2;
        end
        j = 1;
    else
        j = 1;
    end
else
    for i=2:length(rec)
        rec(i).rank_cri = i;
    end
end
ret = rec;
References


VITA

Pannapa Changpetch was born in Bangkok, Thailand. She studied at Chulalongkorn University, receiving a bachelor’s degree in engineering in 1998. She received her master’s degree in economics from Chulalongkorn University in 2001. In the same year, she began work a master’s degree at the Department of Industrial Engineering at the University of Wisconsin–Madison’s and received a master’s degree in industrial engineering in 2003. In 2004, she enrolled at Penn State as a Ph.D. candidate in industrial and manufacturing engineering. However, in 2006, Pannapa changed her research direction and became a Ph.D. candidate in statistics. She graduated with this degree in 2012.

Her research focus is linear models, time series analysis, statistical process control and design of experiments, and data mining and association rules analysis. She has published papers in Quality and Reliability Engineering International and has forthcoming paper in Journal of Computational Statistics and Simulations. Her teaching experience includes statistics and engineering courses ranging from Quality Control and Reliability to Statistical Concepts and Reasoning, as well as online courses for Penn State’s World Campus. In 2011, she received the William L. Harkness Teaching Award for outstanding teaching and other contributions to the Department of Statistics during the 2010 calendar year.