AN IMPROVED ORGANIZATION METHOD FOR ASSOCIATION RULES AND A BASIS FOR COMPARISON OF METHODS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MASOOD JABARNEJAD

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

JUNE 2010

Approval of the thesis:

AN IMPROVED ORGANIZATION METHOD FOR ASSOCIATION RULES AND A BASIS FOR COMPARISON OF METHODS

submitted by **MASOOD JABARNEJAD** in partial fulfillment of the requirement for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Nur Evin Özdemirel
Head of Department, **Industrial Engineering** _____

Prof. Dr. Gülser Köksal
Supervisor, **Industrial Engineering Dept., METU** _____

Assoc. Prof. Dr. Murat Caner Testik
Co-Supervisor, **Industrial Engineering Dept., HACETTEPE** _____

Examining Committee Members

Prof. Dr. Nur Evin Özdemirel
Industrial Engineering Dept., METU _____

Prof. Dr. Gülser Köksal
Industrial Engineering Dept., METU _____

Assoc. Prof. Dr. Murat Caner Testik
Industrial Engineering Dept., HÜ _____

Assoc. Prof. Dr. Yasemin Serin
Industrial Engineering Dept., METU _____

Assist. Prof. Dr. Cem İyigün
Industrial Engineering Dept., METU _____

Date: 04.06.2010

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Masood Jabarnejad

Signature :

ABSTRACT

AN IMPROVED ORGANIZATION METHOD FOR ASSOCIATION RULES AND A BASIS FOR COMPARISON OF METHODS

Jabarnejad, Masood

M.Sc., Department of Industrial Engineering

Supervisor: Prof. Dr. Gülser Köksal

Co-Supervisor: Assoc. Prof. Dr. Murat Caner Testik

June 2010, 86 pages

In large data, set of mined association rules are typically large in number and hard to interpret. Some grouping and pruning methods have been developed to make rules more understandable. In this study, one of these methods is modified to be more effective and more efficient in applications including low thresholds for support or confidence, such as association analysis of product/process quality improvement. Results of experiments on benchmark datasets show that the proposed method groups and prunes more rules.

In the literature, many rule reduction methods, including grouping and pruning methods, have been proposed for different applications. The variety in methods makes it hard to select the right method for applications such those of quality improvement. In this study a novel *performance comparison basis* is introduced to address this problem. It is applied here to compare the improved method to the original one. The introduced basis is tailored for quality data, but is flexible and can be changed to be applicable in other application domains.

Keywords: Data mining, Association rules, Grouping and pruning rules, Comparison of rule reduction methods

ÖZ

BİRLİKTELİK KURALLARI İÇİN İYİLEŞTİRİLMİŞ BİR DÜZENLEME YÖNTEMİ VE YÖNTEMLERİN KARŞILAŞTIRILMASI İÇİN BİR TEMEL

JABARNEJAD, Masood

Yüksek Lisans, Endüstri Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Gülser Köksal

Ortak Tez Yöneticisi: Doç. Dr. Murat Caner Testik

Haziran 2010, 86 Sayfa

Büyük veri tabanlarında, keşfedilmiş birliktelik kurallar kümesi genellikle geniştir ve yorumlaması güçtür. Bu birliktelik kurallarını daha anlaşılır bir hale getirmek için bir kaç gruplandırma ve budama yöntemi geliştirilmiştir. bu yöntemlerden bir tanesi, destek ya da güven ölçüleri için düşük alt sınırlar içeren uygulamalarda (örneğin, ürün/süreç kalitesinin iyileştirilmesi için birliktelik analizi) daha etkili ve daha verimli olacak şekilde iyileştirilmiştir. Kıyaslama veri tabanları üzerindeki deney sonuçları, iyileştirilmiş yöntemin daha fazla kuralı gruplandırdığını ve budadığını göstermektedir.

Literatürde, kuralları gruplandırma ve budama içeren, çok sayıda kural indirgeme yöntemi önerilmiştir. Bu yöntemlerin fazlalığı, kalite iyileştirme gibi uygulamalar için doğru yöntem seçilmesini güçleştirmektedir. Bu problemin çözümü için yeni bir performans karşılaştırma temeli ortaya konulmuştur. Bu temel kullanılarak, iyileştirilmiş yöntemle orijinal yöntem karşılaştırılmıştır. Geliştirilen bu temel asıl olarak kalite verisi için oluşturulmuştur. Ancak, esnek yapısıyla diğer uygulama alanlarında kullanılmak için değiştirilebilir özelliğe sahiptir.

Anahtar kelimeler: Veri madenciliği, birliktelik kuralları, kural gruplandırma ve budama, kural indirgeme yöntemlerinin karşılaştırılması

To my family

ACKNOWLEDGEMENTS

First of all, I would like to give my special thanks to my supervisor Prof. Dr. Gülser Köksal and my co-supervisor Assoc. Prof. Dr. Murat Caner Testik for their continuous support and encouragement throughout this study. I am really thankful for their kindness and patience anytime I needed. Without them, I could not have completed this study. I also would like to thank them for teaching me how to approach and deal with a research work and how to write a scientific paper.

I would like to express my appreciation to the rest of my thesis committee members; Prof. Dr. Nur Evin Özdemirel, Assoc. Prof. Dr. Yasemin Serin, and Assist. Prof. Dr. Cem İyigün for their comments and suggestions.

I am also indebted to my brother for his support and advices. He has always helped me with mathematical concepts whenever needed. He has also been attentive to me every time I wanted to share something.

I am thankful to Assist. Prof. Dr. Abdelaziz Berrado for his kind attention and for the useful information he shared with me.

I am grateful to İlker A. İpekçi and Berna Bakır for their help and for the information they shared with me during the first steps of this study.

I would like to thank to my mother and my father for their endless love, unconditional support, and encouragement to pursue my interests throughout my life.

I would like to thank my friends for their support and help.

TABLE OF CONTENTS

ABSTRACT…..……..……IV

ÖZ.…….……..……V

ACKNOWLEDGEMENTS…...……VIII

TABLE OF CONTENTS…….……IX

LIST OF TABLES..……..……XI

LIST OF FIGURES...……..……XII

CHAPTERS

1. INTRODUCTION……1

2. LITERATURE REVIEW…….7

2.1 Association Rules Background……...…..….7

2.2 Rule Miners……...8

2.3 Interestingness Measures……...8

2.4 Quantitative Association Rules…….9

2.5 Rule Reduction Methodologies……...……10

2.6 Comparing Different Rule Reduction Methods……..13

3. METARULES METHOD…….…….15

3.1 Review of the Method…….15

3.2 Concerns about the Metarules Method…….……...…18

4. THE PROPOSED METHOD……..…...23

4.1 Analysis of Overlap and Containment……....23

4.2 Grouping/Pruning Rules Discovered by Low Confidence Thresholds…...31

5. EXPERIMENTS ON BENCHMARK DATASETS.…….33

5.1 Experiments to Investigate the Volume of Data Scanning …….35

5.2 Experiments to Investigate the Overlap Underestimation and To Compare the Effectiveness in Grouping /Pruning the Rules …….39

6. A NOVEL BASIS TO COMPARE RULE REDUCTION METHODS…....49

6.1 Controlled Experiment……...…….50

6.2 Application……..61

7. CONCLUSION AND FUTURE WORK……..….68

REFERENCES……...71

APPENDICES

A. ASSOCIATION RULES IN THE GENERATED QUALITY DATA……..78

B. CHARTS & ANALYSIS RESULTS FOR EVALUATION MEASURES..79

LIST OF TABLES

TABLES

Table 3.1 Distribution of data with C_1 as the item of C……..……19
Table 3.2 Distribution of data with C_2 as the item of C…..……19
Table 3.3 Discovered rules…….……20
Table 3.4 Discovered metarules……..……21
Table 3.5 Underestimated metarules……...……22
Table 4.1 Significant overlaps discovered by the ODA algorithm ……30
Table 5.1 Datasets used in the experiments ……...……34
Table 5.2 Results for the efficiency of the proposed method in data scanning (fairly low support thresholds)……...……37
Table 5.3 Results for the efficiency of the proposed method in data scanning (lower support thresholds) …….……..……..…38
Table 5.4 Results for Overlap Underestimation (high confidence thresholds)…...…40
Table 5.5 Results for Overlap Underestimation (low confidence thresholds)……41
Table 5.6 Grouping rules using the metarules and the proposed methods (high confidence thresholds)…….……..……...…42
Table 5.7 Grouping rules using the metarules and the proposed methods (low confidence thresholds)…….……..……43
Table 5.8 Pruning rules using the metarules and the proposed methods (high confidence thresholds)…….…..……..……44
Table 5.9 Pruning rules using the metarules and the proposed methods (low confidence thresholds)……...…….……45
Table 5.10 Similarities and differences between sets of pruned rules (low confidence thresholds)…….……..……47
Table 6.1 Expected Logistic function for all combinations of effective process variables …….……53

Table 6.2 The used design for control experiment with two 2-level factors…..……57
Table 6.3 Summary of characteristics and their relations with each other…..…….58
Table 6.4 Summary of the evaluation measures for GP methods…….…….…59
Table 6.5 Collected characteristics and metrics values……..…..……...…62

LIST OF FIGURES

FIGURES

Figure 3.1 Process of mining metarules…..……17
Figure 4.1 Rules and their regions ...…..……24
Figure 4.2 Overlap /containment between rules ……26
Figure 4.3 An algorithm to discover the overlaps between mined rules……...…29
Figure 4.4 Organized rules using metarules...……31
Figure 4.5 Organized rules using overlaps….……32
Figure 4.6 Organized and grouped rules after using overlaps……32
Figure 5.1 Percentages of Overlap Degrees Calculated Without Data Scanning...…39
Figure 5.2 Possible situations between sets of pruned rules……...……46
Figure 6.1: Important failure association rules in sufficiently large generated data...54
Figure 6.2: Performance evaluation process for GP methods…….……56
Figure 6.3 Results of one-way ANOVA on metric M_2……...……63
Figure 6.4 Results of one-way ANOVA on metric M_3...……64
Figure 6.5 Results of one-way ANOVA on metric M_4…...…….65
Figure 6.6 Results of one-way ANOVA on metric M_5……...……66
Figure B.1 Charts & analysis results for metric M_2…….……..79
Figure B.2 Charts & analysis results for metric M_3…..…….……80
Figure B.3 Charts & analysis results for metric M_4…..…….……81
Figure B.4 Charts & analysis results for metric TM_4…...….……82
Figure B.5 Charts & analysis results for metric LM_4 …..…….…..83
Figure B.6 Results of Kruskal-Wallis Test on metric M_4…..…….….84
Figure B.7 Main effects and ANOVA results for metric M_4…..…….85
Figure B.8 Charts & analysis results for metric M_5…..…….……86

CHAPTER 1

INTRODUCTION

In the recent decades, information technology has been advancing dramatically. The rapid improvements in hardware/software devices have enabled markets, business centers, and production units to collect and store relevant data easily and efficiently. Today, all conglomerates, large organizations, manufacturers and even small business companies possess plenty of data reflecting their transactions, operations, or business-relevant activities. Consequently, the complexity and volume of data increase day-to-day. The growth in data complexity requires managers and engineers to be equipped with sophisticated methods to be able to benefit from the valuable knowledge included within the data. On the other hand, the growth in data volume requires designed methods to be efficient and practical in real applications.

Data mining and knowledge discovery involve methods and efforts to address both requirements of sophistication and efficiency [1]. They encompass a variety of techniques which mainly can be classified into four functions: classification, clustering, regression and association rule mining [2]. These functions are not necessarily mutually exclusive and one function can be combined with another to reveal a better quality of extracted knowledge. For example, classification is integrated with association rule mining to deliver more accurate *classifiers* [3] or clustering can be applied to quantitative data to improve the quality of data discretization, which is required by many standard association rule mining algorithms.

One of the most important data mining approaches is the association rule mining which is used to detect hidden affinity patterns in the datasets [4]. Association rules were initially used to study the purchase behaviors of customers in the market. One example for a discovered rule can be, "A customer buying the bread also buys eggs" or in rule format, $\{bread\} \rightarrow \{eggs\}$. This and other discovered rules can help managers to improve their customer services and increase the profitability. However, applications of this approach are not limited to the market basket analysis and can be used by data analysts in different areas.

Advances in sensing and computer technology have enabled companies and manufacturing systems to record many process and product variables. The progressions in collection and retrieval of data, makes most service or manufacturing processes a data-rich environment which is a suitable domain to apply and benefit from data mining techniques including association analysis. In recent decade, Association analysis has been benefited in many real-world applications. For example, Shahbaz et al. [5] apply association rule mining to unearth improvement knowledge for product design and manufacturing process. Another application is introduced by Buddhakulsomsiri et al. [6] where they develop an *association rule-generation* algorithm to identify associations between product features and the occurrence of a particular warranty problem in automotive warranty data.

Association rules can also be used to improve the quality of a product or a manufacturing process by detecting root causes of quality problems and eliminating them. Assume that a manufacturing dataset is comprised of some process variables and a failure variable. The process variables contain different production measures and settings determined and controlled by production engineers. The failure variable contains different kinds of defects in products that occur during production. In such a domain, association analysis can be used to mine interesting rules between the process and failure variables. Interesting rules are usually the rules that indicate hidden and unknown information such that the production engineers are unaware of. Such sort of information can be deployed by engineers to discover hidden root causes of some failures in the system so that the failures can be fixed or eliminated. This

will decrease the defect incidences and hence will increase the quality of manufacturing process or products. Hence, one of the suitable application areas for association rules is the quality domain for discovering the root causes of failures and defects in a manufacturing system.

A successful application of association rules require appropriate initial analysis and settings for the rule mining process, such as appropriate data cleaning, selection of appropriate rule miners and threshold specifications for support and confidence. In general, setting low thresholds is necessary to mine all interesting rules and to prevent information loss. This is because interesting associations are usually observed less frequently. If an association is very prevalent in a system, then it is perceptible and known for human beings there. On the other hand, many important associations can have low confidence, e.g. *failure* associations in a manufacturing system. Failure associations are associations that probably will result in production failure. Let's explain this a little more. Assume there are two unknown failure associations in a production system. The first one has a 100% confidence. This means that whenever the first failure association occurs, then certainly it will cause a failure in the system. The second failure association has just a 50% confidence. This means that if the second failure association occurs many times, then in half of the times a failure will occur but in the other half it will not. On the other hand, let the frequency of second failure association be twice the frequency of first one, i.e. the second one is more frequent. Consequently, both associations can cause almost equal number of failures in the system. However, if a high threshold is specified for the confidence in the initial rule mining step, then the second failure association will not be detected. As a result, to conduct a multifarious failure-diagnosis research, it is necessary to specify a low confidence threshold for rule mining algorithms.

There are many other applications and case studies in which the rules with a fairly low confidence are useful or even more interesting than stronger ones. Coenen et al. [7] discuss how setting the confidence threshold in association rules mining technique can affect the classification applications. They show that because of the nature of some datasets, a low confidence threshold in association analysis is required to reveal the necessary amount of rules to be used in classification process.

Shaw et al. [8] developed some interestingness measures to discover the rules from multi-level datasets. They showed how to discover many low confident but interesting rules and that if in association analysis a high confidence threshold is used, then such important rules will be lost. There are many other real-world problems and applications in which association rules with low confidences contain useful knowledge to be discovered. See for example, [9] for a medical application and [10] for a text mining application. As a matter of fact, efficient and effective organization of rules in such applications might be very beneficial.

Besides various applications of the association rules, there exist some difficulties with association analysis that adversely affect its practical effectiveness in massive datasets. The most important problem with this approach is the overwhelming number of discovered rules [11]. This is because the classic measures of support and confidence generate many redundant rules, especially when low thresholds are set for them, and it is hard to interpret the huge number of mined rules.

In general three kinds of approaches are introduced to reduce the set of mined association rules and make them more understandable. The first approach applies the concept of *closed* set of items [12]. The second approach applies more interestingness measures in addition to support and confidence [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]. Therefore, the set of mined association rules reduces to the set of interesting rules that satisfy support-confidence framework and also some additional interestingness measures. The third approach is based on "similarity" and/or "redundancy" among association rules [11, 17, 27, 28, 29, 30, 31, 32, 33, 34] and it can be categorized into two; grouping and pruning the association rules. Grouping techniques try to summarize the set of discovered rules by clustering "similar" rules. Pruning techniques try to reduce the number of mined or to be mined association rules by detecting "redundant" rules and removing them.

The first approach, mentioned for reducing the large set of discovered association rules, can cause information loss as it is looking for closed frequent itemsets. The second approach evaluates the rules with some pre-specified interestingness

measures. Defining a measure to catch exactly interesting associations can be a very difficult task as the underlying distributions of such associations are unknown. However, the grouping/pruning techniques in the third approach may be helpful as these techniques consider the regions of data and rules together in similarity or redundancy evaluation. Hence, after applying such techniques, the information loss is negligible and the final set of rules is smaller and suitable for later study of data analysts.

The focus of this thesis is on developing an effective rule reduction method particularly for applications requiring low support-confidence thresholds. There are some pre-developed methodologies that try to organize the large sets of mined association rules into human-tractable rule sets. However, many of these approaches may not be appropriate for applications requiring low threshold settings. Berrado and Runger [28] have developed an approach for organizing rules in sparse data. They introduce *metarules* to group and prune association rules. A metarule is a rule between two association rules that are already discovered. In other words, a metarule is an association rule whose antecedent and consequent are two discovered association rules. For convenience throughout this study, this approach is referred to as the metarules method or the metarules approach. Metarules method may encounter some problems in applications including low threshold settings. In this study, we improve their approach to be more efficient in implementation and more effective in grouping and pruning rules especially mined with low threshold settings.

In the literature, many rule reduction methods are proposed for different applications. The variety in methods makes it hard to wisely select the right method for new applications. Hence, developing a basis to evaluate and compare methods and to facilitate this decision is very beneficial. We introduce a novel *performance comparison basis* which enables data analysts to precisely evaluate the performance of different rule reduction methods and then to select the most suitable one. Additionally, we use the presented basis to compare the performance of Berrado and Runger's approach [28] with the one developed here in the quality data domain.

Although the introduced basis is tailored for quality data, it is flexible and can be changed to fit the other subjects and data types.

The organization of the thesis is as follows: In Chapter 2, a background on association rule mining is provided. In Chapter 3, the approach introduced by Berrado and Runger [28] is reviewed in detail and then some concerns related to this approach are discussed with examples. We show how this approach can be more efficient in mining required metarules. Further, we show that this approach may underestimate some significant overlaps, which results in a less effective solution at the following grouping/pruning tasks. In Chapter 4, the overlap and containment of rules are analyzed with new concepts and definitions. Then an algorithm is developed to mine the overlaps in a more efficient way. Experiments on some benchmark datasets are conducted in Chapter 5, where the efficiency and effectiveness of the proposed approach are compared with those of Berrado and Runger [28]. In Chapter 6, we introduce a new basis which enables data analysts to precisely evaluate the performance of different grouping/pruning methods. Then the introduced basis is used to compare the performance of Berrado and Runger's approach [28] with the one proposed here for the quality problems domain. Finally, conclusions and suggestions for future work are presented in Chapter 7.

CHAPTER 2

LITERATURE REVIEW

2.1 Association Rules Background

Association rules are initially introduced in transactional data by Agrawal et al. [4], where each transaction includes some items that are purchased by a specific customer. Hence an association rule is an expression of the form $X \rightarrow Y$ where X and Y are subsets of purchased items with $X \cap Y = \varnothing$. X is referred to as *antecedent* and Y as *consequent* of the association rule (or briefly rule). Furthermore, the union of X and Y is called the *itemset* of the rule. Interestingness of a rule is often measured by its *support* and *confidence*. Support measures the generality of the rule and is equal to the fraction of the transactions that include (or satisfy) both X and Y (rule itemset). Confidence measures the strength (or predictive ability) of the rule and is equal to the fraction of the transactions including X that further include Y. If support and confidence of a rule are above the minimum thresholds, then the rule is discovered (or mined). If the support of an itemset is above the minimum support threshold, then it is called a *frequent* itemset.

In this thesis, some more definitions are considered. It is said that a transaction *supports* or *includes* a rule or that a transaction is *supporter* of a rule if it includes the itemset of that rule. Furthermore, if the itemset of one rule is a proper subset of the itemset of another rule, then the rule with smaller itemset is called the *sub-rule* and the rule with larger itemset is called the *super-rule*.

2.2 Rule Miners

Agrawal et al. [4] introduce the initial algorithms to mine association rules. Based on that work, Agrawal et al. [35] develop the well-known Apriori algorithm which is more efficient. Han et al. [36] propose FP-growth algorithm to mine the frequent patterns. They further develop their algorithm in [37]. Zaki et al. [38] present the concept of *closed* frequent itemsets. An itemset X is called a closed frequent itemset if X is a frequent itemset and further X does not have a superset with as much support as X. They show that the set of closed frequent itemsets is much smaller than the set of frequent itemsets. Later Zaki et al. [39] introduce CHARM algorithm to mine all closed frequent itemsets with the *vertical* data format and then to generate only the non-redundant rules. Also a framework based on closed itemsets is proposed in [12] to drastically reduce the set of mined rules. There are many other developed algorithms to mine association rules such as CLOSET algorithm [40], OPUS algorithm [26, 41]. The correctness and runtime performance of discussed rule miners are evaluated and compared in [42]. Another important rule miner is CARMA, introduced by [43], which compute large itemsets online and typically, it is by an order of magnitude more memory efficient than Apriori. There are other approaches that propose rule miners satisfying specific applications. However, conducting a complete survey on all developed rule miners is beyond the scope of this thesis.

2.3 Interestingness Measures

Various measures and metrics are introduced to evaluate the interestingness or importance of rules. Many of them are fundamentally objective, i.e. they are only based on properties of rules and the data used in mining process. Some other measures are subjective where they also consider the preferences of users who are interested in resulting rules.

There are different objective measures in the literature such as support and confidence [4], gain [17], variance and chi-squared value [22, 23], entropy gain [21,

22], gini [21], laplace [15, 26], lift [14, 16, 18], and conviction [14]. Bayardo et al. [44] show that many of the objective interestingness measures favor the rules that have at least an advantage in support or confidence over other ones, which is called the *sc-optimality*. An important drawback of these measures is that they do not consider the differences in data-regions of rules and therefore direct application of these measures may cause information loss. To prevent this, the sc-optimality is extended to *pc-optimality*, where rules are evaluated with respect to their data-region and confidence. However, the resulting optimal rule set still needs to be grouped and summarized.

Statistical methods are also used to evaluate the interestingness of rules. Brin et al. [13] use the chi-square test to discover correlations between items in a dataset. An algorithm is also developed to mine *correlation rules*. This approach may fail to detect the rare but interesting regularities in datasets as chi-square become an inaccurate interestingness measure when the support of interesting associations is low. For example, it may be ineffective in detecting rare root-causes of quality problems in quality data. Further, the final set of mined rules may contain many overfitting rules and still needs to be reduced. Many other statistical measures and methods are developed to evaluate the rules. In [45] association rules are reviewed from a statistical point of view.

Subjective measures are developed to involve the extra knowledge into interestingness evaluation [19, 20, 24, 25]. Besides the developed measures, selecting the right interestingness measures for evaluation of association rules is addressed in [46]. For more information about different developed interestingness measures, either objective or subjective, readers are referred to [47].

2.4 Quantitative Association Rules

Apriori algorithm requires attributes to have discrete (or categorical) values and not quantitative (or continuous) values. To be able to mine association rules from data including quantitative attributes many discretization techniques are introduced in the

literature [48, 49, 50, 51, 52, 53, 54, 55]. Nevertheless, efforts have been concentrated to develop algorithms to directly mine the quantitative association rules. Srikant et al. [56] introduce a methodology to discover quantitative association rules. In this methodology, quantitative attributes are mapped into partitions and different measures are used to control the degree of partitioning and quality of output association rules. Aumann and Lindell [57] present a new definition of quantitative association rules based on statistical inference theory. This method is for the association rules involving a single discrete or quantitative attribute in the antecedent and a single quantitative attribute in consequent.

2.5 Rule Reduction Methodologies

The most well-known algorithm for association rule mining is the Apriori algorithm [35]. However, effectiveness of the approach may be a concern in some practical settings. The problem is that the number of mined rules is often large [11] and a noticeable amount of these discovered rules are redundant. Redundancy occurs because a large number of rules are overlapped or contained by other rules. Consequently, organizing the discovered rules is a reasonable approach to deal with this problem and it can be categorized into two; pruning and grouping the association rules. Pruning techniques try to reduce the number of mined or to be mined association rules by detecting the redundant rules and removing them. Some pruning techniques are applied during the mining process and are called *constrained-based* rule miners. Others are applied on the already discovered rules. On the other hand, grouping techniques try to summarize the set of discovered rules by clustering the rules, where each cluster consists of the rules arising from roughly or exactly the same regions of data.

Several pruning methods are developed to reduce the redundancy between the rules. Klementin et al. [11] introduce *rule templates* to mine just the rules consistent with user interests or domain knowledge. They construct inheritance hierarchy for attributes and use templates to reduce the discovered rules. Ng et al. [32] give the user more control over association rules mining process and also exploit the user's

feedback to guide the mining process toward the interesting rules discovery. Srikant et al. [33] embed item constraints into association rule mining algorithms to discover rules that the user is interested in.

Bayardo et al. [27] propose constrained-based rule miners to discover association rules satisfying some pre-specified constraints. The pre-specified constraints include support, confidence, and *minimum improvement* constraints. The minimum improvement constraint forces the undesirable super-rules to be pruned. For example, given a super-rule *r*, if confidence of *r* is not considerably more than the largest confidence among the sub-rules of *r*, then *r* is pruned. This constraint will make the rule miner more effective if the dataset is dense. Though, it is a concern when the dataset tends to be sparse. Furthermore, the approach does not group and summarize the discovered rules. Consequently, the overlaps/containments between the discovered rules are unknown and need to be discovered.

An alternative approach is the one proposed in Toivonen et al. [34] where the set of discovered rules is mapped to *rule covers*. A rule cover is a subset of original set of discovered rules with the same consequent. The set of data rows that support the rule cover is equal to the set of data rows that support all discovered rules. Rule covers are obtained in two steps. In the first step, all super-rules are removed since their supporting data rows are subsets of the supporting data rows for their sub-rules. The remaining set is referred as the *structural rule cover*. In the second step, a greedy algorithm is used to reduce the size of structural rule covers further, and to find optimal rule covers. After applying these two steps respectively, the number of discovered rules might be decreased significantly. However, this approach assumes that the discovered association rules are very strong, i.e. all discovered rules have very high confidence values. Nevertheless, the used pruning technique may not be appropriate when the confidence improvement between the discovered rules is considerable since some rules with much higher confidence values will be removed for sake of finding optimal rule covers.

Chawla et al. [29] develop a method for adaptive local pruning of mined association rules using directed hypergraphs. Given a set of mined association rules, the method generates an *Association Rules Network* by a construction algorithm. Then the constructed association rule network provides a graphical method to prune rules by associating redundant rules with hypercycles and reverse hyperedges. The pruning technique is local because it is applied in the context of a goal node. In other words, a rule that is redundant according to a particular goal node may become non-redundant with respect to another goal node. Moreover, when the set of discovered association rules is comprised of the rules with the same consequent, the local pruning technique will not be effective as there will not be any hypercycles or reverse hyperedges in the constructed association rules network. Hence, a global organization of the discovered rules is required.

Some methodologies are developed to find the optimal association rules for numeric attributes [17]. Lent et al. [31] propose an approach to cluster the rules with exactly two numeric attributes in antecedent. They call two rules *adjacent* if their items related to one of the attributes have adjacent values. The adjacent items are combined to form a more general rule. This method is limited to cluster the rules with quantitative items in antecedent and is not applicable when the rules involve categorical values in their left hand side (antecedent).

Different grouping or clustering methods are also introduced. Toivonen et al. [34] use a distance-based clustering approach to group the pre-reduced rules. In this approach, the distance between two rules is defined as the number of data rows that the rules differ. This grouping approach organizes the rules more. However, as the rules get larger, that is their support increases, their relative distance with other rules will get larger as well. Consequently, the grouping task may be adversely affected by support value. Some approaches are also developed to resolve this problem [30]. However, these distance-based clustering methods are sensitive to the asymmetric relationship between the data regions of rules and this drawback is shown in [28].

On the other hand, Berrado et al. [28] develops an approach which is not based on distance-based clustering. In this approach, metarules are used to group and prune the overlap-involved rules. Metarules are the rules that imply associations between the discovered rules themselves. When all metarules are discovered, two rules are called *equivalent* rules if they satisfy the following two conditions: first, they appear with together in two metarules; second, they appear with other rules in other metarules in the same way. Clustering equivalent rules is the grouping technique used in this approach. This grouping technique summarizes the discovered rules noticeably. Furthermore, it is not sensitive to either the largeness of rules' region or the asymmetric relationship between the data regions of rules. However, when the data tend to be dense and the set of discovered rules include many rules with low confidence, the effectiveness of this technique may be reduced; i.e. it may not be able to group all eligible rules. This happens because many of the metarules remain undiscovered and therefore many eligible rules are not considered as equivalent. Further details about it are provided in Chapter 3. In addition to that, the approach will require overwhelming number of data scans to mine required metarules when the set of mined rules is large. Berrado et al. [28] uses metarules to prune equivalent rules. In this approach, after rules are grouped, the super-rules are pruned in the sets of equivalent rules. Clearly the pruning effectiveness of this approach depends on the results of the previous grouping step. There is not much work in the literature to address the association rules mining problem in applications including low threshold settings. In this thesis, we modify the metarules method to be more efficient and more effective in grouping/pruning rules mined with low support or confidence thresholds.

2.6 Comparing Different Rule Reduction Methods

Different approaches are developed for rule grouping/pruning tasks. However, their practicality may vary in different applications and for various data types. Some of them are more effective when the target data is dense [27] and some are more effective when the data is sparse [28]. Others may deal with the set of discovered rules better only when it includes strong rules i.e. the rules with high confidence [30,

34]. In addition to that, different pruning methods introduce different definitions for redundant rules [11, 29, 32, 33, 34]. This is expected as the definition of redundant rules can differ from one application to another. Because of the variety in methods and applications, it is necessary to select the right method for the right application. In [28] the groups formed using the metarules method is compared with the clusters formed using two different distance-based grouping techniques. The comparison is made group by group and case by case by analyzing illustrative figures. However, it is very hard for data analysts to use illustrative figures to compare many rule groups one by one. Further, there are other important criteria to be considered in methods evaluation such as information loss prevention and redundancy removal. We are unaware of any developed general model for evaluation of rule reduction methods. Hence, developing a basis to evaluate and compare the performance of various methods in a target application is very beneficial. We address this problem by introducing a new basis by which data analysts can evaluate the performance of different rule reduction methods and then select the best one.

CHAPTER 3

METARULES METHOD

3.1 Review of the Method

Organizing association rules to a manageable size is an active research area since interpretation of many discovered rules is a difficult task. To group and prune association rules, Berrado and Runger [28] propose metarules method and show how the sparseness of data can cause redundancies among the discovered rules. Metarules method is based on overlaps/containments between the discovered rules having the same consequent.

Let $T = \{t_1, t_2, \ldots, t_n\}$ be a set of transactions and $R = \{r_1, r_2, \ldots, r_m\}$ be a set of discovered rules with the same consequent mined from dataset T. First, the approach maps the set of transactions T to another set $Q = \{q_1, q_2, \ldots, q_l\}$ where $l \leq n$ such that every element q_j of Q is a subset of rules in R such that:

$$q_j = \{r_i \in R \mid t_j \text{ includes antecedent of } r_i\}$$

In other words, every element q in Q includes the rules from R that their antecedents are supported by the corresponding transaction t in T. For convenience, in the next chapters, we will refer to Q as *Q-set*. The Apriori algorithm [35] is applied on the Q-set to find the associations between the discovered rules. The approach confines Apriori algorithm to mine only the rules including exactly one item (here rule) in antecedent and one item (here rule) in consequent. The support threshold is suggested to be set to 0% and the confidence threshold to be set to a high value such as 90% or more. Note that this confidence threshold is for mining metarules. It measures the strength of metarules and is different from the confidence threshold

already used for mining initial rules. The resulting rules are referred to as the metarules, which represent the overlap/containment between the original discovered rules. To illustrate the approach consider the following sample dataset with 4 transactions.

t_1	a, b, c, d
t_2	a, b, d
t_3	c, d
t_4	c, e

Let us concentrate on the rules with a minimum support of 40% and a minimum confidence of 50%. Also assume that the consequent of rules is confined to the item $\{d\}$. Hence, rules r_1:$\{a\}\rightarrow\{d\}$, r_2:$\{b\}\rightarrow\{d\}$, r_3:$\{c\}\rightarrow\{d\}$, and r_4:$\{a, b\}\rightarrow\{d\}$ are discovered. When rules are discovered, the approach maps the example dataset with 4 transactions to the Q-set which is like the following:

q_1	r_1, r_2, r_3, r_4
q_2	r_1, r_2, r_4
q_3	r_3
q_4	r_3

Then the approach reapplies Apriori algorithm on this Q-set with a 0% support threshold value and a high confidence threshold value. Here, assume the threshold for confidence is set to 100%. Note that this confidence threshold (100%) is for mining metarules and is different from the confidence threshold of 50% which is already used for mining initial rules. Hence, the following steps are taken by Apriori algorithm: First, frequent itemsets with one item (or 1-itemsets [35]) are mined. Here, candidate 1-itemsets are $\{r_1\}$, $\{r_2\}$, $\{r_3\}$, and $\{r_4\}$. As the support threshold is set to zero, all of these candidate 1-itemsets are considered as frequent. Then candidate itemsets with 2 items (or 2-itemsets) are considered and their supports are calculated. Here, candidate 2-itemsets are $\{r_1, r_2\}$, $\{r_1, r_3\}$, $\{r_1, r_4\}$, $\{r_2, r_3\}$, $\{r_2, r_4\}$,

and $\{r_3, r_4\}$. Again for the same reason, all of these 2-itemsets are considered frequent. As the Apriori algorithm is confined to just mine association rules (here metarules) with exactly one item (here rule) in both antecedent and consequent, it does not go beyond 2-itemsets. Finally, the algorithm generates all strong association rules (here metarules) by considering frequent 2-itemsets. The process on the example Q-set is depicted in Figure 3.1.

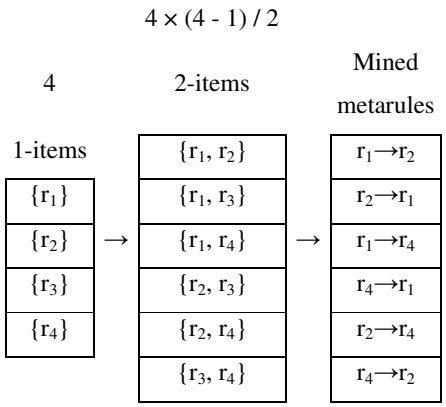


Figure 3.1 Process of mining metarules

As illustrated in Figure 3.1, Apriori algorithm requires many scans to the Q-set to calculate the supports for 1-itemsets and 2-itemsets. For 1-itemsets, it requires 4 scans, as there are 4 discovered rules, and for 2-itemsets it requires $4 \times (4 - 1) / 2 = 6$ scans. Therefore, the data in Q-set is scanned $4 + 4 \times (4 - 1) / 2 = 10$ times in order to mine all metarules. Hence in general case, if there are m discovered rules, then the constructed Q-set will be scanned $m + m \times (m - 1) / 2$ times in order to mine all metarules.

After discovering all metarules, two rules are called *equivalent* if they mutually contribute together in two metarules and further, their antecedent or consequent contributions in other metarules with other rules are the same. In the illustrated

example, the rules r_1 and r_2 are equivalent because: First, they mutually contribute together in two metarules of $r_1 \rightarrow r_2$ and $r_2 \rightarrow r_1$. Second, both rules contribute as antecedent in metarules $r_1 \rightarrow r_4$ and $r_2 \rightarrow r_4$ with r_4 and again both rules contribute as consequent in metarules $r_4 \rightarrow r_1$ and $r_4 \rightarrow r_2$ with r_4. Grouping these equivalent rules is the technique used in the grouping task. In the illustrated example the rules r_1, r_2, and r_4 are grouped. For pruning, the super-rules in the sets of equivalent rules are pruned. The equivalent super-rules are named *complex* rules. In the example, r_4 is a complex rule as it is an equivalent super-rule. Hence, this rule is pruned.

3.2 Concerns about the Metarules Method

In mining association rules, setting a low threshold for support can reveal almost all regularities in datasets. However, it will result in drastically increased number of discovered rules. In this case, the metarules method would be computationally demanding as it reapplies Apriori algorithm [35] to mine initial metarules. Just assume there are 1000 discovered rules. To mine all metarules, the Apriori algorithm will have to take $1000 + 1000 \times (1000 - 1) / 2 = 500{,}500$ references back to Q-set to calculate the confidence of all candidate metarules. This can be a challenge when the datasets are large such as the ones associated with manufacturing systems. Even in some very large datasets, the metarules method seems to be impossible to be implemented.

Metarules method mines overlaps between already discovered association rules and then uses them to group or prune the rules. Its effectiveness in grouping and pruning seems to increase when the data distribution is sparse. However, there may be some concerns related to its effectiveness when data are not purely sparse or that the initial confidence thresholds in mining association rules are low. This is because the approach defines overlaps between rules as the intersections of antecedents of rules. When data are not sparse or that low confidence thresholds are set, then the method may underestimate some of the significant intersections of itemsets of rules and consequently not discover the corresponding metarules. Hence, some of the

candidate metarules can be neglected and not reported to the data analyst. This partial inaccuracy in metarules mining, would adversely affect the effectiveness of the approach in following grouping and pruning steps; particularly in the applications including many rules with low confidence. Association rules with low confidence are used in many real applications [7, 8, 9, 10]. Furthermore, as discussed in the introduction chapter, in mining association rules from quality data one needs to set low support-confidence thresholds to prevent information loss.

The foregoing discussions about metarules method are explained and illustrated in an example tabular dataset in the following. Assume *A*, *B*, and *C* are three categorical attributes. The attribute *A* has two items A_1 and A_2, *B* has two items B_1 and B_2 and *C* has two items C_1 and C_2. The distribution of data is shown in Tables 3.1 and 3.2.

Table 3.1 Distribution of data with C_1 as the item of *C*

	B_1	B_2
A_1	50	0
A_2	50	0

Table 3.2 Distribution of data with C_2 as the item of *C*

	B_1	B_2
A_1	20	20
A_2	20	20

In these tables, each cell number indicates the number of data rows that include the corresponding items. For example in Table 3.1, the number 50 in the cell related to items A_1 and B_1 means there are fifty data rows that include items A_1, B_1, and C_1. In general a dataset can be dense, sparse, or dense in some regions while sparse in the others. For preserving this generality, we considered both dense and sparse data regions in the illustration. The data in Table 3.1 tend to represent a sample sparse region while the data in Table 3.2 tend to represent a sample dense region. Assume it

is required to mine the association rules that include items from the attributes *A* or *B* as antecedent and item C_1 from the attribute *C* as consequent. Also assume that the thresholds for support and confidence are specified as 10% and 50%, respectively. Apriori algorithm is used to mine association rules which are summarized in Table 3.3.

Table 3.3 Discovered rules

Rule ID	Rule	Sup. (%)	Conf. (%)
r_1	$\{A_1\}\rightarrow\{C_1\}$	27.8	55.6
r_2	$\{A_2\}\rightarrow\{C_1\}$	27.8	55.6
r_3	$\{B_1\}\rightarrow\{C_1\}$	55.6	71.4
r_4	$\{A_1,B_1\}\rightarrow\{C_1\}$	27.8	71.4
r_5	$\{A_2,B_1\}\rightarrow\{C_1\}$	27.8	71.4

Now consider that the metarules method is applied to the data in Tables 3.1 and 3.2 and to the rules in Table 3.3 for organizing the rules. After mapping data in Tables 3.1 and 3.2 to the Q-set, Apriori algorithm is reapplied to discover the metarules. Hence, Apriori algorithm considers all pairs of rules as candidate metarules. However, noticeable amount of rules explain distinct regions of data and are neither contained nor overlapped by each other. To explain this more, assume candidate metarules $r_1 \rightarrow r_2$ and $r_2 \rightarrow r_1$ in the example above. A_1 and A_2 are the antecedents of the rules r_1 and r_2 respectively. We know that none of the data rows can include both items A_1 and A_2 at the same time because the data is in tabular format and these items belong to the same attribute *A*. Hence, the confidence of $r_1 \rightarrow r_2$ and $r_2 \rightarrow r_1$ is equal to zero. In a sample set of rules, there may be many of such redundant candidate metarules that can be removed at first without spending time on them. Therefore, the comparison task between them for overlap calculations can be relaxed. Now assume that during applying metarules method, the specified threshold for the confidence of metarules is set to 100% to reveal the containments between the rules. Again note that this confidence threshold (100%) is for mining metarules and is different from the confidence threshold of 50% which is already used in mining rules in Table 3.3.

After metarules is applied to the data in Tables 3.1 and 3.2 and to the rules in Table 3.3, the mined metarules are illustrated in Table 3.4.

Table 3.4 Discovered metarules

Metarule ID	Metarule	Conf. (%)
mr_1	$r_4 \rightarrow r_1$	100
mr_2	$r_4 \rightarrow r_3$	100
mr_3	$r_5 \rightarrow r_2$	100
mr_4	$r_5 \rightarrow r_3$	100

The discovered metarules in Table 3.4 do not reflect all containments between the rules in Table 3.3. Let us explain one of them. According to the data in Table 3.1, there are 50 data rows that support the rule r_1. These data rows are the same data rows that support the rule r_4. There are not other data rows that exclusively support one of rules r_1 or r_4. Hence both rules r_1 and r_4 are mutually contained in each other. In Table 3.4, the discovered metarule mr_1: $r_4 \rightarrow r_1$ reflects the containment of r_4 in r_1. However, none of discovered metarules in Table 3.4 reflect the containment of r_1 in r_4. The latter containment can be reflected by the metarule $r_1 \rightarrow r_4$. However this metaulre is not discovered. This is because the confidence of metarule $r_1 \rightarrow r_4$ is calculated as 77.8% which is below the threshold 100%. We conclude that the confidence of this metarule is underestimated. In this example, there are totally four metarules whose confidences are underestimated. They are listed in Table 3.5. All metarules in this table reflect containments between the rules that are not reflected by discovered metarules in Table 3.4.

Table 3.5 Undiscovered metarules

Metarule ID	Metarule	Confidence (%)
mr_5	$r_1 \rightarrow r_4$	77.8
mr_6	$r_1 \rightarrow r_3$	77.8
mr_7	$r_2 \rightarrow r_5$	77.8
mr_8	$r_2 \rightarrow r_3$	77.8

Note that in this example, the underestimation happens when antecedent of metarules include rules with low confidences. The antecedents of all undiscovered metarules in Table 3.5 include one of the rules r_1 or r_2 which have the lowest confidence among all discovered rules in Table 3.3. The metarules method assumes that data is very sparse. It accepts a transaction as supporter of a rule if the transaction includes just the rule antecedent. However, when data is not very sparse, the antecedent-including transactions for a rule can differ from the itemset-including ones for that rule. Hence, some confidences for metarules can be underestimated and this may results in less discovered metarules. This can be addressed by slightly changing the definition of confidence for the metarule. To address the discussed concerns and problems, in the next chapter we modify the metarules method by introducing a new method. The proposed method requires less data scans in the process of discovering overlaps /containments between rules. Further, the proposed method prevents the underestimation of overlaps or containments. This prevention enables the proposed method to group and prune more rules.

CHAPTER 4

THE PROPOSED METHOD

4.1 Analysis of Overlap and Containment

In this chapter, some basic concepts are defined and used to discuss and discover all possible overlap/containment cases that can occur between association rules. Assume $A = \{A_1, A_2, \ldots, A_N\}$ is a set of N categorical attributes such that each attribute A_i includes some discrete values. Assume $V(A_i)$ represents the set of all discrete values belonging to attribute A_i. Here, an item is defined as an attribute-value pair as (A_i,a) where $a \in V(A_i)$. The set $\{A_i\}\times\{V(A_i)\}$ is the set of all items for attribute A_i and therefore the set $I = \bigcup_1^N \{A_i\}\times\{V(A_i)\}$ is the set of all items for all attributes. Also assume a tabular dataset $D = \{d_1, d_2, \ldots, d_N\}$ comprised of N data rows such that each data row includes exactly one item from each attribute. Let $R = \{r_1, r_2, \ldots, r_m\}$ be the set of m discovered association rules from D. For simplicity we call a subset of data rows a *region*[1]. Then we conceive an association rule as a region of data rows, where the region consists of the supporter data rows of that association rule. A definition of a rule region is given below:

> **Definition 4.1:** Given an association rule $r:\{X\}\rightarrow\{Y\}$, the *region* h of the rule r is the set of all data rows that support the rule r. Equivalently, $h = \{d \in D \mid d$ include both antecedent and consequent of $r\}$.

[1] We have borrowed the term of *region* from [58].

To make the concept of a rule region clear, an illustration is used in the following. Consider the sample dataset and also the regions for 4 discovered association rules of r_1:{(A, a_1)}→{(C, c_1)}, r_2:{(A, a_2)}→{(C, c_1)}, r_3:{(B, b_2)}→{(C, c_1)}, and r_4:{(A, a_1), (B, b_2)}→{(C, c_1)}, shown in Figures 4.1.

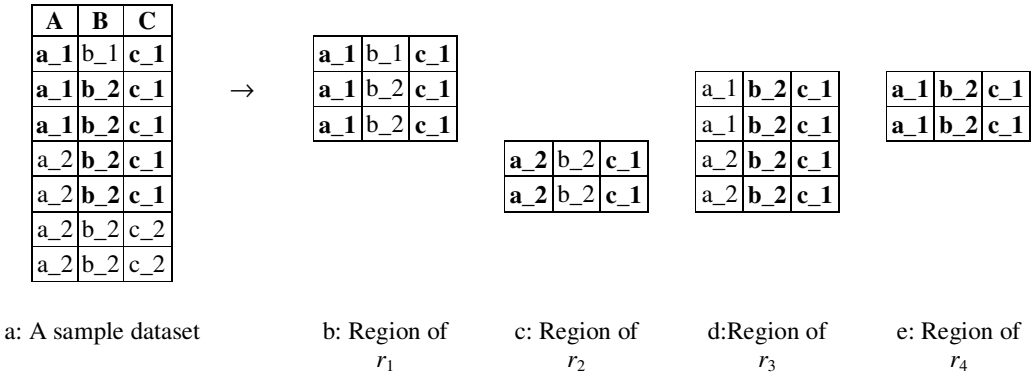


Figure 4.1 Rules and their regions

In Figure 4.1, regions of rules r_1 and r_2 are disjoint. The same relation is true about regions of rules r_2 and r_4. On the other hand, the regions of rules r_1 and r_3 are intersecting. Regions of rules r_2 and r_4 are subsets of the region of the rule r_3. Also the region of the rule r_4 is a subset of the region of the rule r_1.

Hence, regions of different association rules can be disjoint from, intersected by or be subsets of each other. Let r_1 and r_2 be two association rules with regions h_1 and h_2, respectively. If regions h_1 and h_2 are intersecting, then we say that there is an *overlap* between r_1 and r_2. We also call the intersection of regions h_1 and h_2 as the *joint-region*. Hence, the joint-region is the set $h_1 \cap h_2$. Assume α percent of data rows included in h_1 are also included in h_2. Here α is called the *overlap degree* in r_1 and (n($h_1 \cap h_2$) / n(h_1)) × 100 is its value[1]. The *containment* among rules is a specific case of overlap in which the overlap degree is equal to 100%. To evaluate the

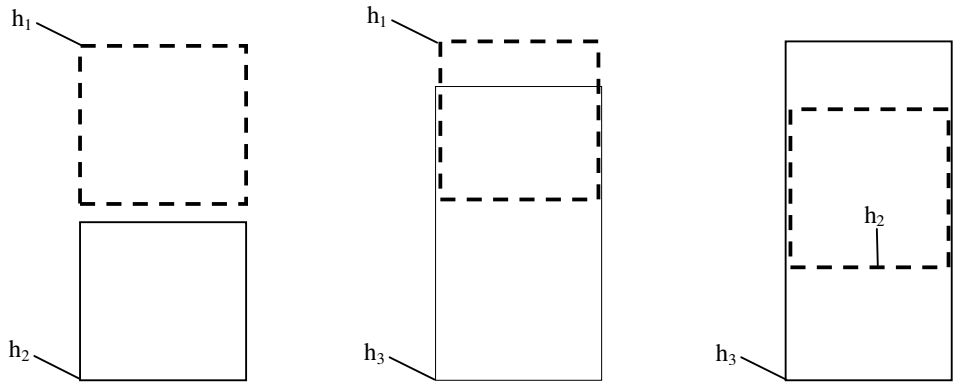
[1] Throughout this thesis, the notation n(•) denotes number of elements in a set. For example if A is a set, then n(A) denotes number of elements in the set A.

significance of overlaps between rules, a *minimum overlap* threshold is set. Any overlap with the degree above minimum threshold is considered as significant.

> **Definition 4.2**: We say that the rule r_1 is *significantly overlapped* by the rule r_2 if the inequality $(n(h_1 \cap h_2) / n(h_1)) \times 100 \geq mo$ is satisfied, where h_1 and h_2 are regions of rules r_1 and r_2, respectively and mo is a minimum threshold that belongs to [0, 100].

If rules r_1 and r_2 satisfy the Definition 4.2, then this relation is denoted by $r_1 \rightarrow r_2$. This is the same notation used in metarules method [28]. If both $r_1 \rightarrow r_2$ and $r_2 \rightarrow r_1$ are satisfied, then we say that the rules r_1 and r_2 are *mutually* and *significantly* overlapped.

Let us analyze overlaps and containments between the example rules in Figure 4.1. Also assume that a minimum overlap threshold of 60% is considered. In this figure, there is an overlap between r_1 and r_3. The overlap degree in r_1 is $(2 / 3) \times 100 = 66.6\%$ and in r_3 is $(2 / 4) \times 100 = 50\%$. Hence, r_1 is significantly overlapped by r_3 which can be denoted by $r_1 \rightarrow r_3$, but r_1 does not significantly overlap r_3. There is not any overlap between r_1 and r_2. The same thing is true about rules r_2 and r_4. Further, r_4 is contained in both rules of r_1 and r_3 (equivalently $r_4 \rightarrow r_1$ and $r_4 \rightarrow r_3$). The rule r_2 is only contained in the rule r_3 (or $r_2 \rightarrow r_3$). To illustrate these concepts more, overlaps / containments between example rules of r_1, r_2, and r_3 in Figure 4.1 are depicted in Figure 4.2.



(a): no overlap between r_1 and r_2

(b): overlap between r_1 and r_3

(c): containment between r_2 and r_3

Figure 4.2 Overlap /containment between rules

In the metarules method, first the Q-set[1] is constructed and then Apriori algorithm is reapplied on it to mine metarules. This idea is used here to mine the overlaps between the rules. Here, we construct Q-set in a slightly different way. We map the set of data rows D to the new set $Q = \{q_1, q_2, \ldots, q_l\}$ where $l \leq n$ such that every element q_j of Q is the set of rules data row d_j supports, that is:

$$q_j = \{r_i \in R \mid d_j \text{ includes antecedent and consequent of } r_i\}$$

In other words, every element q in Q includes the rules from R that their itemsets are supported by the corresponding data row d in D. To prevent confusion with the metarules constructed Q-set, let the new constructed set be denoted as *NQ-set*. After constructing the NQ-set, similarly we reapply the Apriori algorithm on it. This time Apriori would mine overlaps instead of metarules. Hence, the overlap underestimation of the metarules method is rectified. However, if the number of mined rules is large, then Apriori algorithm will scan NQ-set several times in the process of discovering significant overlaps. To address this disadvantage as well, an algorithm is developed, but before going through it, the overlap between rules is analyzed more in the following.

[1] The set on which Apriori algorithm is applied to mine the metarules. For more details see Chapter 3.

To calculate overlap degrees between rules r_1 and r_2, obviously the number of data rows in their joint-region (i.e. $n(h_1 \cap h_2)$) is required. Note that in the constructed NQ-set, $n(h_1 \cap h_2)$ is equal to the number of elements in NQ-set that include both rules r_1 and r_2. However, counting these elements in NQ-set can be a time-consuming task, especially when the NQ-set includes very large number of elements. Fortunately, there are two cases in which the value of $n(h_1 \cap h_2)$ does not have to be counted and can be calculated directly.

The first case is when both rules have items from the same attributes. If there is one item in the antecedent of the first rule and another item in the antecedent of the second rule, where both items are from the same data attribute, then the joint-region of r_1 and r_2 has to be empty (i.e. $h_1 \cap h_2 = \varnothing$). This is because the data is in tabular format and a data row cannot simultaneously include two items from an attribute. For example, consider the rules r_1:{(A, a_1)}→{(C, c_1)} and r_2:{(A, a_2)}→{(C, c_1)} in Figure 4.1. Both of these rules have an item from the attribute A. Obviously, there is not any data row in Figure 4.1 which include both items of (A, a_1) and (A, a_2). For simplicity, we refer to this case as the *same-attribute* case. Later in experiments in Chapter 5 we will show that many rule comparisons in overlap discovery process satisfy the same-attribute case; especially when the support threshold in initial rule mining step is set to a low value.

The second case is when one of the rules is a super-rule of another one. In this case, the joint-region of rules is actually the region of the super-rule. Hence, the overlap degree in the super-rule is 100% i.e. the super-rule is contained in the sub-rule. Further, the overlap degree in the sub-rule is equal to the support of the super-rule divided by the support of the sub-rule. For example, consider the rules r_3:{(B, b_2)}→{(C, c_1)}, and r_4:{(A, a_1), (B, b_2)}→{(C, c_1)}, in Figure 4.1. The rule r_4 is a super-rule of the rule r_3. Hence, the overlap degree in r_4 is 100%. Also the overlap degree in r_3 is equal to (28.5 / 57.0) × 100 = 50%. For simplicity, we refer to this case as the *super-rule* case. This case is stated and proved in the following;

Lemma: If the rule r_1 is a sub-rule of the rule r_2, then the overlap degree in r_1 is equal to $[s(r_2) / s(r_1)] \times 100$, where $s(r)$ denotes the support of the rule r.

Proof: Assume that D denotes the original dataset and h_1 and h_2 are regions of rules r_1 and r_2, respectively. We know that $s(r_1) = n(h_1) / n(D)$ and $s(r_2) = n(h_2) / n(D)$. Hence, we have: $[s(r_2) / s(r_1)] \times 100 = [(n(h_2) / n(D)) / (n(h_1) / n(D))] \times 100 = (n(h_2) / n(h_1)) \times 100$. As r_2 is a super-rule of r_1, therefore $h_2 = h_1 \cap h_2$ is satisfied. Hence, we have: $(n(h_2) / n(h_1)) \times 100 = (n(h_1 \cap h_2) / n(h_1)) \times 100$, which is equal to the overlap degree in r_1. ■

The lemma above implies that already discovered super-rules are another opportunity in directly calculation of overlap degrees in sub-rules. Based on the explained cases, an overlap discovery algorithm (*ODA*) is developed here, in which pair-wise comparison is used to discover all overlap/containment between the discovered rules. Hence, if there are m discovered rules then the algorithm considers $m \times (m - 1) / 2$ pairs of rules. For each rule pair, if the same-attribute case is satisfied, then overlap degrees in both rules are set to zero. Further, for each rule pair, if the super-rule case is satisfied and not the same-attribute case, then the overlap degree in the super-rule is set to 100% and the overlap degree in the sub-rule is directly calculated using the support of the super-rule. The developed algorithm is given in Figure 4.3.

In Chapter 3, we explain that the metarules method reapplies Apriori algorithm on the Q-set. We have also showed that the Apriori algorithm scans the Q-set m times for 1-itemsets and $m \times (m - 1) / 2$ times for 2-itemsets where m is the number of initially discovered rules. Note that an x-itemset means a set with x items. The set of all 2-itemsets in the Apriori algorithm is equal to the set of all rule pairs considered in the ODA algorithm. Here, an important point is that the ODA algorithm does not scan the NQ-set when the same attribute or the super-rule cases appears. If the difference in the largeness of the Q- and NQ- sets is negligible, then applying ODA

algorithm is more efficient than reapplying Apriori algorithm from the aspect of data scans number. If in an experiment there is not any same-attribute or super-rule cases, then still ODA algorithm scans data less than the Apriori algorithm, because the Apriori algorithm requires additional m scans for 1–itemsets which the ODA algorithm does not require. We do not calculate the computational complexities of the ODA algorithm and the approach of reapplying the Apriori algorithm used in the metarules method. It requires future work and here we just show how the ODA algorithm has the potential to mine the overlaps more efficiently.

Overlap Discovery Algorithm (ODA)

Input: Discovered association rules $R = \{r_1, r_2, \ldots, r_m\}$
Dataset $D = \{d_1, d_2, \ldots, d_N\}$
NQ-set $Q = \{q_1, q_2, \ldots, q_L\}$
Itemsets $I(r)$ of the rule $r \in R$
Supporter elements $SE(rs) = \{q_j \in Q \mid rs \subseteq q_j\}$ for the rule(s) $rs \subseteq R$
Overlap threshold *min_ovlp*
Output: Set of significant overlaps O
Method:

```
while n(R) > 1 do
  choose one rule from R and assign it to r_i;
  for all r_j ∈ R − {r_i} do
    if same-attribute case is satisfied then do // check the same-attribute case
      overlap = 0;
    else do
      A =: I(r_i) ∪ I(r_j);                  // A is the union of the itemsets of both rules
      if A = I(r_i) or A = I(r_j) then do  // check the super-rule case
        s_k =: min{s(r_i),s(r_j)};
      else do
        s_k =: n( SE({r_i,r_i}) ) / N;
      end;
      // s_k is equal to the number of data rows in the joint-region divided by N
      d_i =: s_k / s(r_i);          // calculate the overlap degree in r_i
      d_j =: s_k / s(r_j);          // calculate the overlap degree in r_j
      if d_i ≥ min_ovlp then do     // check the overlap significance
        O =: O ∪ {r_i → r_j}; // add the discovered significant overlap to the O set
      end;
      if d_j ≥ min_ovlp then do     // check the overlap significance
        O =: O ∪ {r_j → r_i}; // add the discovered overlap to the O set
      end;
    end;
  end;
  R = R − {r_i};  // remove r_i from R
end;
```

Figure 4.3 An algorithm to discover the overlaps between mined rules

Here, there is another important point. In Chapter 3, it is shown that the metarules method may underestimate some significant overlaps between the discovered rules and hence, may not discover all of them. As the ODA algorithm mines overlaps instead of metarules, it also prevents the overlap underestimations.

The ODA is applied on the rules in Table 3.3 in Chapter 3 with a threshold of 100% for minimum overlap. The discovered significant overlaps are reported in Table 4.1. The resulting overlaps include both discovered metarules in Table 3.4 and underestimated ones in Table 3.5 in Chapter 3. Therefore, the underestimation is prevented. In this example there are totally 10 pairs of rules to be considered for the NQ-set scanning and the overlap analysis. In 4 comparisons, the same-attribute case and in other 4 comparisons, the super-rule case is satisfied. In the remaining 2 comparisons, the algorithm scans the NQ-set to count the number of data rows in related joint-regions. As a result, the ODA scans the NQ-set only 2 times. If the metarules method was applied, then it would scan the Q-set 10 times, which is 5 times more than the number of data scans of the ODA algorithm.

Table 4.1 Significant overlaps discovered by the ODA algorithm

Overlap ID	Overlap	Overlap Degree
o_1	$r_1 \rightarrow r_4$	100
o_2	$r_4 \rightarrow r_1$	100
o_3	$r_1 \rightarrow r_3$	100
o_4	$r_4 \rightarrow r_3$	100
o_5	$r_2 \rightarrow r_5$	100
o_6	$r_5 \rightarrow r_2$	100
o_7	$r_2 \rightarrow r_3$	100
o_8	$r_5 \rightarrow r_3$	100

4.2 Grouping/Pruning Rules Discovered by Low Confidence Thresholds

Grouping (or clustering) the association rules is an important technique to summarize the discovered rules into more interpretable clusters where each cluster comprises the interrelated rules. Different approaches are developed for the rule grouping task. However, their practicality may vary in different applications and various data types. Some of them are more effective when the data is dense and some are more effective when the data is sparse. Others may deal with the set of discovered rules better only when it includes strong rules (the rules with high confidence). Here, our intention is to effectively group the rules mined by low confidence threshold. We use the same rule-grouping idea used in Berrado et al. [28]. The only difference is that we group rules using the discovered overlaps between the rules rather than using metarules. In other words, we group two rules if they are mutually and significantly overlapped, and also they significantly overlap (or are significantly overlapped by) other rules in the same way. By doing that, the underestimation will be prevented and all large overlaps will be mined and as a result, all equivalency-deserving rules will be grouped. To show the advantage of our method we compare the grouping rules using metarules with the grouping rules using overlaps by the example introduced in Chapter 3. Both, metarules in Table 3.4 and overlaps in Table 4.1 are used to group the discovered rules in Table 3.3. The resulting rule-organizations by the metarules method and the proposed approach are depicted in Figures 4.4 and 4.5, respectively.

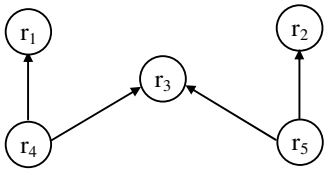


Figure 4.4 Organized rules using metarules

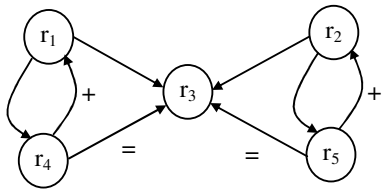


Figure 4.5 Organized rules using overlaps

In Figure 4.4, the illustration as considered in Berrado et al. [28] is used where a modification of that is presented in Figure 4.5. In Figure 4.5, besides the representation of overlaps among the rules, the confidences of sub- and super-rules are compared with each other and the results are reflected by the corresponding mathematical symbols. For example consider the rules r_3 and r_4 in Figure 4.5. The equality symbol '=' on the directed overlap arc $r_4 \rightarrow r_3$ means that confidence of super-rule r_4 is equal to confidence of sub-rule r_3. In the case of a higher /lower confidence, the plus /minus symbol is used. By doing this, the super-rules, having predictive strength not more than the sub-rules, can be distinguished easily and then can be pruned in the case of requirement. Similarly, the super-rules with more confidence can be preserved from pruning. In Figure 4.4, there is not any equivalency between rules. Therefore, rules are not grouped together. Nevertheless, in Figure 4.5 there are 2 pairs of equivalent rules. The rules r_1 and r_4 are equivalent. The same is true about rules r_2 and r_5. As a result, the rules in Figure 4.5 are summarized more in Figure 4.6. Furthermore, the rules r_4 and r_5 are equivalent super-rules with no confidence advantage over sub-rule r_3 and can be pruned. Hence, in this example, 5 mined rules are reduced to 3 rules by using overlaps. The considered example is just for the illustration purpose. In the next section, we will compare the effectiveness of metarules method and our method in grouping /pruning rules when they are applied on real benchmark datasets.

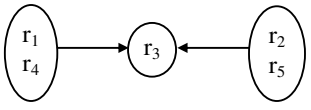


Figure 4.6 Organized and grouped rules after using overlaps

CHAPTER 5

EXPERIMENTS ON BENCHMARK DATASETS

In this chapter some experiments are conducted to compare the efficiency, accuracy and effectiveness of the proposed method with the metarules method. In the experiments, some benchmark datasets are considered which are obtained from the UC Irvine machine learning repository [59] and are listed below;

- Iris Plants Database (Iris)
- Johns Hopkins University Ionosphere Data (Ion)
- Statlog Project Heart Disease Data (Heart)
- Mushroom Data (Mush)
- Thyroid Disease Data (Thyroid),
- Hayes-Roth Data Set (Hayes)
- Nursery Database (Nursery)

The characteristics of datasets are summarized in Table 5.1. Each dataset includes a number of data rows (or records). In these datasets, each data row consists of some discrete or continuous items allowed to be considered as rule antecedents. Furthermore, the datasets include a few class labels to be considered as rules consequent.

Table 5.1 Datasets used in the experiments

Dataset	Data rows	Attributes	Numeric Attributes	Classes
Iris	150	4	4	3
Ion	351	34	32	2
Heart	270	13	6	2
Mush	8,124	22	0	2
Thyroid	7,200	21	6	3
Hayes	132	5	0	3
Nursery	12,960	8	0	5

Apriori algorithm implemented by Christian Borgelt [60, 61] is used to mine class association rules. Note that the Apriori algorithm requires all itemsets to be categorical. On the other hand, the discretization of the continuous items may adversely affect the quality of discovered association rules. However, we do not explore this issue here and use the naïve equal frequency discretization method with four bins. To discretize the continuous attributes, the tools in SPSS Clementine 11.1 [62, 63, 64] are used. In mining the initial association rules, Apriori algorithm is confined to consider at most four items for antecedent of rules except for experiments including *recommend* and *very_recom* classes of Nursery data. The item number constraint for these two classes is relaxed because a few rules or no rules are mined if the item number constraint is applied.

Our intention is to compare the two approaches at applications involving low-threshold settings for support or confidence. Hence, we run two different sets of experiments each one with different intentions. The first set of experiments is related to method efficiency investigation when the support threshold decreases. The criterion for the method efficiency is the percentage of overlaps whose degrees are calculated without scanning the data. Note that here the computational complexity of the data scanning is assumed to be high. Hence here, the less a method scans the data in the process of overlap /containment discovery, the more efficient it is. In Chapter 3, it is shown that the Apriori algorithm, reapplied in the metarules method, scans the data for all possible overlap /containment cases. On the other hand, in Chapter 4 it is explained that our developed ODA algorithm is able to directly calculate overlap

degrees without data scanning when one of the following two cases occurs. One case is the same-attribute case and it occurs when the two rules, in a rule pair, have items from same attributes. Another case is the super-rule case and it occurs when one of the rules, in the rule pair under comparison, is a super-rule of another rule. Hence, we would like to see how many times the same-attribute case and the super-rule case occurs. The more these two cases are faced, the less the ODA algorithm scans the data and therefore, the more efficient it is in comparison to the Apriori algorithm reapplied in the metarules method. As a result, we just apply our approach in the first set of experiments and then reckon the percents of overlap degrees calculated with and without data scanning.

The second set of experiments has two intentions. The first intention is related to investigating the overlap underestimation when the confidence threshold in rule mining process decreases. In Chapter 3, it is shown that the metarules method may underestimate some confidences of candidate metarules and do not reflect the related overlaps or containments. In these experiments, we aim to compare both the metarules and our methods in mining significant overlaps. The criterion in comparing the methods is the number of underestimated significant overlaps which is equal to the difference between the number of mined metarules and the number of mined significant overlaps. Hence, the more this difference is, the more organization the proposed method do on rules than do the metarules method. The second intention in these experiments is to investigate the effectiveness of both approaches in grouping /pruning of the rules. The criterion for the method effectiveness is the number of constructed rule clusters after grouping equivalent rules and also the number of remaining rules after pruning the complex rules. The definitions of equivalent and complex rules are reviewed in Chapter 3. Therefore, in the second set of experiments both approaches are applied.

5.1 Experiments to Investigate the Volume of Data Scanning

In the first set of experiments at the beginning, Apriori algorithm is applied twice on each dataset to mine two sets of association rules. For the first one, a fairly low

support threshold or briefly FLST is set. For the second one, a lower support threshold or briefly LST is used. Please note that different thresholds are used for different class labels and data sets since the class labels are not distributed evenly on the data cases. After applying Apriori algorithm, the NQ-set for each set of rules is constructed. Then the ODA algorithm is applied on two sets of discovered rules and NQ-sets. In all applications of ODA algorithm, a minimum overlap threshold of 100% is set. The results for the experiments involving FLST settings are summarized in Table 5.2. Also the results for the experiments involving LST settings are summarized in Table 5.3. First two columns in Tables 5.2 and 5.3 are related to datasets' name and the target class labels. The third column shows the threshold adjustments for support and confidence. The fourth column indicates the number of mined association rules. The fifth column shows the number of all rule pairs under comparison. Actually, the values in the fifth column are calculated by the values in the fourth column. For example, assume the first experiment in Table 5.2 for the *setosa* class in the Iris dataset. The value of fourth column is 51. The value of fifth column is 1,275 which is calculated by $51 \times (51 - 1) / 2$. It means that 1,275 rule pairs will be compared and analyzed for the overlap discovery. The sixth column shows the percent of rule pairs whose rules satisfy the same attribute case and hence, the data scanning is prevented. Note that for simplicity the same attribute case is briefly referred to as *Case* 1. Similarly, the seventh column shows the percent of rule pairs whose rules satisfy the super-rule case and therefore, the data scanning is prevented. For simplicity, the super-rule case is briefly referred to as *Case* 2. The eighth column is actually the sum of percentages in the sixth and seventh columns. The percentages in this column indicate the total percent of rule pairs in which overlap degrees are calculated without any data scanning. The last column reflects the percent of rule pairs in which overlap degrees are actually minied by scanning the data. This is briefly referred to as *Mined* in the head of ninth column in Tables 5.2 and 5.3

In Table 5.2 in dataset Iris for the class *setosa*, 80% of overlap degrees are directly calculated and only 20% of overlap degrees are calculated by data scanning. In Table 5.2, the best result is obtained from the dataset Nursery for the class *priority* in which

in all comparisons, Case 1 occurs. Therefore, the data is never scanned for this data class. The worst result belongs to dataset Hayes for *class* (3) in which the data is scanned for calculations of all overlap degrees.

Table 5.2 Results for the efficiency of the proposed method in data scanning (fairly low support thresholds)

Dataset	**Class**	**Sup./Conf. (%)**	**Rules (#)**	**Comparison (#)**	**Case 1 (%)**	**Case 2 (%)**	**Cases 1,2 (%)**	**Mined (%)**
Iris	Setosa	1/95	51	1,275	65.3	14.7	80.0	20.0
	Versicolor	1/95	52	1,326	81.4	6.7	88.1	11.9
	Virginica	1/95	45	990	65.1	15.8	80.9	19.1
Ion	Good	10/95	1,153	664,128	32.4	0.4	32.8	67.2
	Bad	10/95	30	435	8.7	5.5	14.2	85.8
Heart	Absence	10/80	346	59,685	22.3	2.0	24.3	75.7
	Presence	10/90	73	2,628	1.0	3.9	4.9	95.1
Mush	Edible	20/95	528	139,128	0.0	1.9	1.9	98.1
	Poisonous	20/95	846	357,435	0.0	1.0	1.0	99.0
Thyroid	Class (1)	0.01/95	174	15,051	36.2	0.9	37.1	62.9
	Class (2)	0.01/95	704	247,456	53.5	0.2	53.7	46.3
	Class (3)	25/95	533	141,778	1.2	2.0	3.2	96.8
Hayes	Class (1)	5/85	6	15	60.0	20.0	80.0	20.0
	Class (2)	5/85	6	15	60.0	20.0	80.0	20.0
	Class (3)	5/85	3	3	0.0	0.0	0.0	100
Nursery	not_recom	10/85	12	66	15.1	16.7	31.8	68.2
	recommend	0.007/50	5	10	40.0	40.0	80.0	20.0
	very_recom	0.2/50	11	55	67.3	0..0	67.3	32.7
	priority	2/85	5	10	100.0	0.0	100.0	0.0
	spec_prior	5/85	4	6	16.6	16.7	33.3	66.7

In Table 5.3, the support thresholds are set to lower values and the same experiments are conducted for second time. In the experiments illustrated in Table 5.3, the best result is obtained from Iris dataset for class *versicolor* in which at near 91% of comparisons, one of the cases 1 or 2 occurs. The worst result belongs to dataset Thyroid for *class* (3) in which 85% of overlap degrees are calculated by data scanning.

Table 5.3 Results for the efficiency of the proposed method in data scanning (lower support thresholds)

Dataset	Class	Sup./Conf. (%)	Rules (#)	Comparison (#)	Case 1 (%)	Case 2 (%)	Cases 1,2 (%)	Mined (%)
Iris	Setosa	0.1/95	71	2485	72.7	10.7	83.4	16.6
	Versicolor	0.1/95	72	2556	85.3	5.6	90.9	9.1
	Virginica	0.1/95	63	1953	76.5	10.9	87.4	12.6
Ion	Good	6/95	11,212	62,848,866	34.0	0.1	34.1	65.9
	Bad	6/95	887	392,941	16.1	0.5	16.6	83.4
Heart	Absence	1/95	4,452	9,907,926	63.1	0.1	63.2	36.8
	Presence	1/95	3,588	6,435,078	59.4	0.1	59.5	40.5
Mush	Edible	10/95	2,752	3,785,376	19.0	0.3	19.3	80.7
	Poisonous	10/95	5,083	12,915,903	18.6	0.2	18.8	81.2
Thyroid	Class (1)	0.001/95	174	15,051	36.2	0.9	37.1	62.9
	Class (2)	0.001/95	704	247,456	53.5	0.2	53.7	46.3
	Class (3)	20/95	3,997	7,986,006	14.7	0.3	15.0	85.0
Hayes	Class (1)	1/85	65	2,080	78.9	4.7	83.6	16.4
	Class (2)	1/85	60	1,770	77.6	5.5	83.1	16.9
	Class (3)	1/85	85	3,570	58.7	5.2	63.9	36.0
Nursery	not_recom	1/85	616	189,420	50.6	1.9	52.5	47.5
	recommend	0.0007/50	5	10	40.0	40.0	80	20.0
	very_recom	0.1/50	53	1,378	79.5	2.5	82	18.0
	priority	1/85	23	253	81.0	4.8	85.8	14.2
	spec_prior	1/85	114	6,441	58.9	3.2	62.1	37.9

The results for directly calculated overlap degrees for all datasets classes are summarized in Figure 5.1. For each dataset class and each FLST and LST setting, the percent of candidate overlaps whose degrees are directly calculated by Case 1 or Case 2 are provided. In 9 out of 20 experiments involving FLST setting, more than half of overlap degrees are directly calculated by just considering Cases 1 or 2. On the other hand, in 14 out of 20 experiments with LST setting, more than half of overlap degrees are directly calculated by just considering cases 1 or 2. This means that in 14 experiments, the ODA algorithm scans data less than the Apriori algorithm reapplied in the metarules method. This implies that the efficiency of ODA algorithm increases when the support threshold decreases. For example, in dataset Nursery for the class spec_prior, 33.3% of candidate overlaps belong to case 1 or 2 when rules are mined with 5% support threshold. For the same class, percentage of candidate overlaps belonging to case 1 or 2 increases to 62.1% when rules are mined with a lower 1% support threshold.

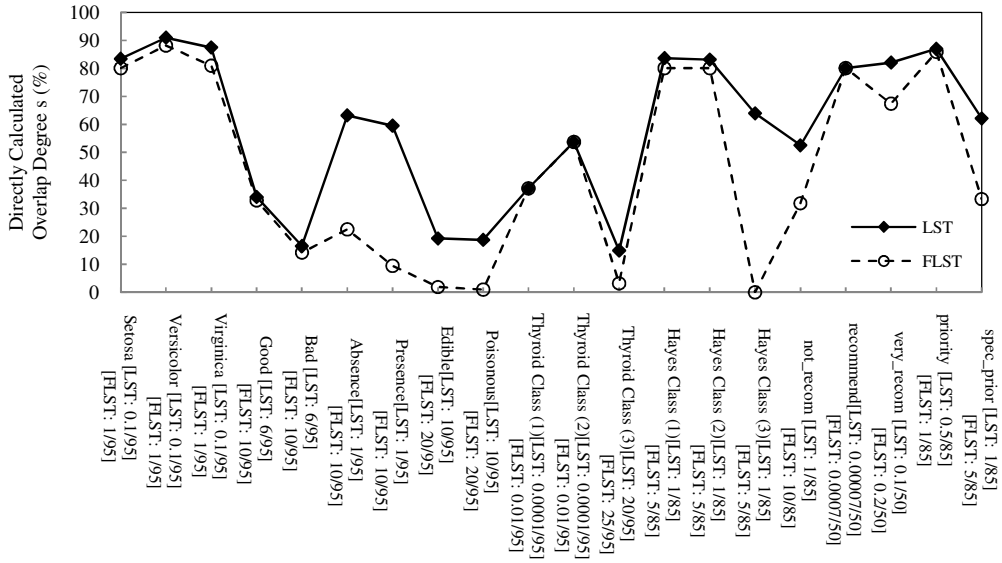


Figure 5.1 Percentages of Overlap Degrees Calculated Without Data Scanning

5.2 Experiments to Investigate the Overlap Underestimation and To Compare the Effectiveness in Grouping /Pruning the Rules

As discussed before in Chapter 3, the metarules method may underestimate some significant overlaps, especially when the set of rules include many rules with low confidence. In the second set of experiments, we explore this more. Afterwards, we use the mined metarules and overlaps to show how the metarules and proposed methods group /prune rules and then compare the effectiveness of the methods. In these experiments, Apriori algorithm is applied two times on each dataset. For the first time, a high confidence threshold is set which we refer to as HCT setting. For the second time, a lower confidence threshold is used which we refer to as LCT setting. Please note that different thresholds are used for different class labels and data sets since the class labels are not distributed evenly on the data cases. After applying the Apriori algorithm and constructing the Q-sets and the NQ-sets, both the ODA algorithm and the metarules method are applied on all discovered rules. For the ODA algorithm a minimum overlap threshold of 100% is set and for the Apriori algorithm reapplied in the metarules method, a minimum confidence threshold of

100% and a minimum support threshold of 0% are set. Note that this 100% confidence threshold setting in the metarules discovery is different from the HCT and LCT settings used in mining the initial association rules. The results for the experiments involving HCT settings are summarized in Table 5.4. Also the results for the experiments involving LCT settings are summarized in Table 5.5.

In 5 out of 20 experiments shown in Table 5.4, the numbers of discovered overlaps and metarules are different from each other. In the Mush dataset, there are 6,262 underestimated overlaps for the class *Edible*, and 24,513 underestimated overlaps for the class *Poisonous*. Also in the Ion dataset with the class *Good*, discovered overlaps are *25%* more than discovered metarules, and in the Nursery dataset for class *recommend*, mined overlaps are twice the mined metarules

Table 5.4 Results for Overlap Underestimation (high confidence thresholds)

Dataset	**Class**	**sup/conf (%)**	**Rules (#)**	**Metarules (#)**	**Overlaps (#)**
Iris	Setosa	1/95	51	241	241
	Versicolor	1/95	52	164	164
	Virginica	1/95	45	261	261
Ion	Good	10/95	1,153	11,248	14,153
	Bad	10/95	30	42	42
Heart	Absence	10/80	346	1,246	1,246
	Presence	10/90	73	107	110
Mush	Edible	20/95	528	85,273	91,535
	Poisonous	20/95	846	364,522	389,035
Thyroid	Class (1)	0.01/95	174	8,914	8,914
	Class (2)	0.01/95	704	44,676	44,676
	Class (3)	25/95	533	5,170	5,170
Hayes	Class (1)	1/85	65	185	185
	Class (2)	1/85	60	170	170
	Class (3)	1/85	85	230	230
Nursery	not_recom	1/85	616	3,185	3,185
	recommend	0.007/50	5	4	8
	very_recom	0.1/50	53	34	34
	priority	1/85	23	12	12
	spec_prior	1/85	114	204	204

Table 5.5 Results for Overlap Underestimation (low confidence thresholds)

Dataset	**Class**	**sup/conf (%)**	**Rules (#)**	**Metarules** (#)	**Overlaps** (#)
Iris	Setosa	1/30	65	398	442
	Versicolor	1/30	98	589	685
	Virginica	1/30	74	444	490
Ion	Good	10/50	3,215	68,741	107,486
	Bad	10/50	174	790	790
Heart	Absence	10/50	514	3,555	3,561
	Presence	10/50	226	1,282	1,317
Mush	Edible	20/85	998	175,102	273,786
	Poisonous	20/85	1,035	462,530	536,501
Thyroid	Class (1)	0.01/50	538	39,999	60,746
	Class (2)	0.01/55	1,041	80,936	88,016
	Class (3)	25/50	2,928	53,000	53,000
Hayes	Class (1)	1/30	283	2,245	3,203
	Class (2)	1/30	285	2,262	3,443
	Class (3)	1/30	97	250	255
Nursery	not_recom	1/30	1,231	8,940	12,740
	recommend	0.007/30	12	15	62
	very_recom	0.1/30	245	415	567
	priority	1/30	1,086	5,709	5,709
	spec_prior	1/30	960	4,865	4,865

In Table 5.5, the differences have increased. This time, in 16 out of 20 conducted experiments shown in Table 5.5, the numbers of discovered overlaps and metarules are different from each other. This implies that the overlap underestimation in experiments with LCT settings is more than the one in experiments with HCT settings. This time in the Mush dataset, there are 98,684 underestimated significant overlaps for the class *Edible*, and 73,971 underestimated significant overlaps for the class *Poisonous*. In dataset Nursery in the class *recommend*, the number of mined significant overlaps is more than 4 times bigger than the number of mined metarules. There are other observed large differences. For example, in Ion dataset for the class *Good*, in Thyroid dataset for the class (1) and in Hayes dataset for the class (2), the numbers of mined significant overlaps are more than 1.5 times larger than the numbers of mined metarules. As a result, in more than half of the conducted experiments with LCT settings, the metarules method underestimates many significant overlaps. These significant overlaps are preserved in the proposed

approach. This inaccuracy in the metarules method results in less effectiveness in the succeeding steps of the method, i.e. in grouping and pruning the rules. On the other hand, as our method preserves all significant overlaps, it is capable to group or prune all deserved rules. Now, we are going to show this by using both methods to group and prune the discovered rules. Let's first start with the grouping task. The results for grouping the rules in Table 5.4 are illustrated in Table 5.6 and the results for grouping the rules in Table 5.5 are summarized in Table 5.7.

Table 5.6 Grouping rules using the metarules and the proposed methods (high confidence thresholds)

Dataset	**Class**	**sup/conf (%)**	**Rules (#)**	**Simplified Rules (#) (Metarules Approach)**	**Simplified Rules (#) (Proposed Approach)**
Iris	Setosa	1/95	51	41	41
	Versicolor	1/95	52	34	34
	Virginica	1/95	45	30	30
Ion	Good	10/95	1,153	504	424
	Bad	10/95	30	15	15
Heart	Absence	10/80	346	340	340
	Presence	10/90	73	72	70
Mush	Edible	20/95	528	93	82
	Poisonous	20/95	846	27	21
Thyroid	Class (1)	0.01/95	174	20	20
	Class (2)	0.01/95	704	62	62
	Class (3)	25/95	533	409	409
Hayes	Class (1)	1/85	65	46	46
	Class (2)	1/85	60	31	31
	Class (3)	1/85	85	73	73
Nursery	not_recom	1/85	616	616	616
	recommend	0.007/50	5	5	3
	very_recom	0.1/50	53	53	53
	priority	1/85	23	23	23
	spec_prior	1/85	114	114	114

Table 5.7 Grouping rules using the metarules and the proposed methods (low confidence thresholds)

Dataset	Class	sup/conf (%)	Rules (#)	Simplified Rules (#) (Metarules Approach)	Simplified Rules (#) (Proposed Approach)
Iris	Setosa	1/30	65	55	47
	Versicolor	1/30	98	80	69
	Virginica	1/30	74	58	51
Ion	Good	10/50	3,215	1,414	841
	Bad	10/50	174	87	87
Heart	Absence	10/50	514	508	506
	Presence	10/50	226	225	222
Mush	Edible	20/85	998	181	132
	Poisonous	20/85	1,035	52	29
Thyroid	Class (1)	0.01/50	538	144	70
	Class (2)	0.01/55	1,041	225	161
	Class (3)	25/50	2,928	2,532	2,532
Hayes	Class (1)	1/30	283	242	155
	Class (2)	1/30	285	252	150
	Class (3)	1/30	97	85	83
Nursery	not_recom	1/30	1,231	1,231	616
	recommend	0.007/30	12	12	3
	very_recom	0.1/30	245	245	233
	priority	1/30	1,086	1,086	1,086
	spec_prior	1/30	960	960	960

The first four columns in Tables 5.6 and 5.7 are the same as previous tables. The fifth column indicates the number of simplified rules using discovered metarules. Number of simplified rules is equal to the number of rule groups and the number of ungrouped rules. The sixth column shows the number of simplified rules using discovered overlaps. In Table 5.6, in 15 experiments, out of 20, both approaches group rules equally. However, in the other 5 experiments the proposed approach groups more rules. The strength of our approach in rule grouping task significantly increases when the initial confidence thresholds for mining the rules decrease. This is reflected in the results summarized in Table 5.7 which include experiments with LCT settings. In majority of experiments in Table 5.7, proposed approach simplifies the rules more. For example in the Iris dataset for the class *setosa*, the number of simplified rules by using metarules reduces to 55 rules. In the proposed approach 8 more rules are grouped and the number of simplified rules reduces to 47. For classes

of *priority* and *spec_prior* in the dataset Nursery, for the class *Bad* in Ion dataset and for the class (3) in the Thyroid dataset, the proposed approach does not simplify more effectively. However for other classes and datasets, it does. For example, the simplified rules in *not_recom* and *recommend* classes, when the proposed approach is applied, are respectively half and one fourth of the simplified rules when the metarules is applied.

After grouping rules, the pruning step is applied by both methods to reduce the discovered rules in Tables 5.4 and 5.5. Complex rules are pruned between equivalent rules, resulting from the metarules method and the proposed approach. The results after pruning the complex rules in Table 5.6 are illustrated in Table 5.8 and the results after pruning the complex rules in Table 5.7 are summarized in Table 5.9.

Table 5.8 Pruning rules using the metarules and the proposed methods (high confidence thresholds)

Dataset	**Class**	**sup/conf** (%)	**Rules** (#)	**Remaining Rules (#) (Metarules Approach)**	**Remaining Rules (#) (Proposed Approach)**
Iris	Setosa	1/95	51	42	42
	Versicolor	1/95	52	38	38
	Virginica	1/95	45	33	33
Ion	Good	10/95	1,153	544	486
	Bad	10/95	30	15	15
Heart	Absence	10/80	346	340	340
	Presence	10/90	73	72	70
Mush	Edible	20/95	528	162	152
	Poisonous	20/95	846	75	62
Thyroid	Class (1)	0.01/95	174	60	60
	Class (2)	0.01/95	704	284	284
	Class (3)	25/95	533	409	409
Hayes	Class (1)	1/85	65	52	52
	Class (2)	1/85	60	48	48
	Class (3)	1/85	85	78	78
Nursery	not_recom	1/85	616	616	616
	recommend	0.007/50	5	5	3
	very_recom	0.1/50	53	53	53
	priority	1/85	23	23	23
	spec_prior	1/85	114	114	114

Just like the previous results, in Table 5.8, in 15 experiments out of 20, both approaches prune rules equally. However, in the other 5 experiments, the proposed approach prunes more rules. As was the case in rule grouping, the strength of our approach in rule reduction significantly increases when the initial confidence thresholds for mining the rules decrease. It is shown in Table 5.9, where in 16 out of 20 experiments, the proposed approach prunes more rules. For example, in the Mushroom dataset for the class *Poisonous* and in the Nursery dataset for the class *not_recommend*, our method prunes, respectively around 50% and 55% more rules. Another interesting observation belongs to Ion dataset for the class *Good*. For this data class, the metarules method reduces the discovered rules from 3,215 rules into 1,461 rules. For the same data class, our method reduces the initial discovered rules into just 856 rules.

Table 5.9 Pruning rules using the metarules and the proposed methods (low confidence thresholds)

Dataset	**Class**	**sup/conf** (%)	**Rules** (#)	**Remaining Rules (#) (Metarules Approach)**	**Remaining Rules (#) (Proposed Approach)**
Iris	Setosa	1/30	65	56	47
	Versicolor	1/30	98	84	78
	Virginica	1/30	74	61	54
Ion	Good	10/50	3,215	1,461	856
	Bad	10/50	174	87	87
Heart	Absence	10/50	514	508	506
	Presence	10/50	226	225	222
Mush	Edible	20/85	998	306	255
	Poisonous	20/85	1,035	104	58
Thyroid	Class (1)	0.01/50	538	219	180
	Class (2)	0.01/55	1,041	456	424
	Class (3)	25/50	2,928	2,532	2,532
Hayes	Class (1)	1/30	283	252	180
	Class (2)	1/30	285	259	185
	Class (3)	1/30	97	90	88
Nursery	not_recom	1/30	1,231	1,231	620
	recommend	0.007/30	12	12	9
	very_recom	0.1/30	245	245	235
	priority	1/30	1,086	1,086	1,086
	spec_prior	1/30	960	960	960

The results in Table 5.9 show that the proposed approach prunes more rules. However, we have not checked what the remaining rules are. For each data class in Table 5.9, there are two sets of remaining rules. One set is obtained by the metarules method and the other one is obtained by our approach. Here, one can ask how much these two sets of the remaining rules are consistent with each other. Assume P_1 denotes the set of pruned rules after applying the metarules method and P_2 denotes the set of pruned rules after applying our approach. Then, the question is which of the situations in Figure 5.2 occur for each dataset experiment.

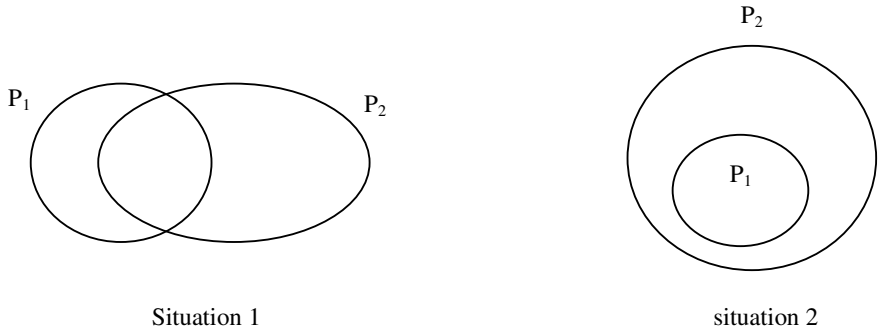


Figure 5.2 Possible situations between sets of pruned rules

In situation 1, we have $P_1 \cap P_2 \neq \varnothing$. If situation 1 is the case, then it means that we have pruned some rules that the metarules method have not and vice versa. In situation 2, we have $P_1 \subset P_2$. If situation 2 is the case, then it means that we have pruned all rules that the metarules method has done and furthermore, we have pruned exclusively some more rules that the metarules method has not. Situation 2 may be acceptable in many applications, but the same inference cannot be easily reached when situation 1 happens. To address this concern, two studies are undertaken. The first one is to collect the set of remaining rules in all experiments in Table 5.9, investigate them and see which of the situations occurs for which datasets. The second study is to conduct a *Controlled Experiment* to compare the performance of both approaches in a systematical way. The last study is the subject of the next chapter and we explain it there.

Regarding the first study, we consider all pruned rules in experiments of Table 5.9, and then investigate the similarities and differences between the set of pruned rules by our method and that by the metarules method. The results are illustrated in Table 5.10.

Table 5.10 Similarities and differences between sets of pruned rules (low confidence thresholds)

Dataset	**Class**	**Common**	**Exclusively Pruned (#) (Metarules Approach)**	**Exclusively Pruned (#) (Proposed Approach)**
Iris	Setosa	9	0	9
	Versicolor	14	0	6
	Virginica	13	0	7
Ion	Good	1,754	0	605
	Bad	87	0	0
Heart	Absence	6	0	2
	Presence	1	0	3
Mush	Edible	692	0	51
	Poisonous	931	0	46
Thyroid	Class (1)	319	0	39
	Class (2)	585	0	32
	Class (3)	396	0	0
Hayes	Class (1)	31	0	72
	Class (2)	26	0	74
	Class (3)	7	0	2
Nursery	not_recom	0	0	611
	recommend	0	0	3
	very_recom	0	0	10
	priority	0	0	0
	spec_prior	0	0	0

Again the first two columns in Table 5.10 are the dataset description where the other columns are related to the pruned rules. The third column represents the number of rules, which are pruned by both methods. The fourth column represents the number of rules exclusively pruned just by the metarules method and the fifth column indicates the number of rules exclusively pruned just by our method. Hence, in Table 5.10, the sum of the third and the fourth columns is equal to the number of rules

pruned by the metarules method for that experiment. Similarly, in each experiment the sum of the third and the fifth columns is equal to the number of rules pruned by our method for that experiment. Fortunately in all experiments, Situation 2 is the case. Hence, we conclude that in all conducted experiments the proposed method was able to prune all rules that did the metarules method and in addition to that, it was able to exclusively prune more rules.

In summary, we have applied both the metarules method and our method on some benchmark datasets to summarize and reduce the amount of initially discovered rules. First, the efficiency of the methods is investigated from the aspect of data scanning volume. Results show that when the initial support threshold in the rule mining process is set to a low value, then the proposed method scans the data less than the metarules method. Second, the accuracy of the metarules method in the overlap mining process is investigated. It is shown that when the initial confidence threshold in the rule mining process is set to a low value, the metarules method underestimates many significant overlaps. These significant overlaps are preserved in the proposed approach and this makes the proposed method capable to group or prune more rules. At the last, the set of pruned rules by both methods are analyzed to check the consistence between them. Fortunately, in all experiments the proposed method is able to prune all rules that did the metarules method and in addition to that, it is able to exclusively prune more rules.

CHAPTER 6

A NOVEL BASIS TO COMPARE RULE REDUCTION METHODS

The results of experiments on some benchmark data (Chapter 5) shows that in almost all datasets the proposed approach groups and prunes more rules. However, pruning more rules by an approach does not necessarily indicate its better performance. To explain this, let *A* and *B* be two different methods to group and prune association rules. Also assume the method *A* prunes more rules than *B* method. Here, it is not clear that the exclusively pruned rules by method *A* are all redundant rules, i.e. *A* method may prune some non-redundant rules and cause information loss. On the other hand, it is also possible that *B* method exclusively prunes some redundant rules that *A* method is not able to prune. As a result, these issues should be considered when the performances of different methods are compared. In general, we can consider four criteria in evaluating performance of grouping/pruning methods: grouping (or clustering) strength, pruning strength, information loss and computational complexity. The general intention of the methodologies is to maximize the first and second criteria and simultaneously minimize the third and fourth ones. In this chapter, we introduce a novel comparison basis, which enables data analysts to precisely evaluate the performance of different rule reduction methods. In the introduced basis, first, second and third method-evaluation criteria are considered. Although the introduced basis is tailored for quality data, it is flexible and can be changed to fit other contexts and data types. For convenience, we will call association rules grouping and pruning methods briefly the *GP* methods. We will also use the terms of rule reduction methods and GP methods with the same meaning.

6.1 CONTROLLED EXPERIMENT

The intention of this chapter is to design and conduct a new series of experiments by which the performance of different GP methods can be measured and compared. We call this new series of experiments the *controlled experiment* as it is a set of somehow controlled experiments to measure the performance of different methods. The idea is to intentionally generate some data with known rules and then apply different rule reduction methods on the rules mined from the generated data. First, we explain the way the datasets are generated. The inspiration of this work is rule mining for quality improvement. Hence, we generate an artificial data within this context and call it the *quality* data. To this end, data are supposed to include some independent and random variables x_1, x_2, x_3, …, x_k representing manufacturing process variables and one binary variable, z representing process/product failure status. Here, we use the well-known logistic regression model [65] to predict the probability of not observing the failure event. In logistic regression, a measure of the total contribution of all independent predictors x_1, x_2, x_3, …, x_k and error used in the model is defined as the following:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_k x_k + \varepsilon \qquad (6.1)$$

and it is known as the logit. Here, the predictors are assumed to be k effective process variables on the failure event in the system. The term ε in (6.1) denotes the error that cannot be accounted for by the effective process variables. It can be considered as the effect of uncontrollable factors such as environmental conditions, limitations in human carefulness and etc. The error is assumed to have normal distribution and the model is assumed to simulate the failure incidence of a manufacturing system. According to the type of the simulated system, predictors can be assigned suitable probability distributions. The parameter β_0 is the *intercept*, which here represents the initial effect of system on potential failures. Here, it can be interpreted as the effect of initial setups required before production. The parameters

β_1, β_2, β_3, …, β_k are regression coefficients of process variables x_1, x_2, x_3, …, x_k, respectively. Given the logit, the logistic function is defined as the following:

$$f(y) = \frac{e^y}{e^y + 1} \qquad (6.2)$$

Therefore, the output $f(y)$ is confined to values between 0 and 1. Here, this output can be interpreted as the probability of not observing a failure. The failure status can be determined by the output values of the logistic function as follows:

$$z = \begin{cases} 1 & \text{if } f(y) \le 0.5 \\ 0 & \text{if } f(y) > 0.5 \end{cases} \qquad (6.3)$$

Hence, if the total contribution of all process variables, i.e. the logit equation in (6.1) results in an output probability of 0.5 or less in the logistic function (6.2), then we conclude that a failure occurs in the manufacturing system and the failure status variable takes a value of 1 or $z = 1$. On the other hand, if that probability is greater than 0.5, then we conclude that no failure occurs (success occurs) and $z = 0$.

Here, we assume there are three process variables which have significant effects on the failure event. Let us call them *effective* process variables. They are represented by predictors x_1, x_2 and x_3. The predictors are independent of each other and have the underlying discrete uniform distribution with possible values of 0 and 1. When $x_i = 1$, it indicates that the i^{th} effective process variable is active in the system and when $x_i = 0$ it indicates that the i^{th} effective process variable is inactive in the system. They are also assumed to have equally negative effects on logit y and consequently positive effects on failure status z. Therefore, coefficients β_1, β_2 and β_3 are set to −1. The initial effect of setup β_0 on logit y is set to +1.999. The term ε in the logit function (6.4) denotes the error that cannot be accounted for by the process variables. It can be considered as the effect of uncontrollable factors such as environmental conditions, limitations in human carefulness and etc. The error is assumed to have normal

distribution with 0 mean and 0.1 variance; in other words, $\varepsilon \sim N(0,0.1)$. Therefore, the logit equation in (6.1) becomes:

$$y = 1.999 - x_1 - x_2 - x_3 + \varepsilon \qquad (6.4)$$

In summary, the process of quality data generation is as following. First the variables x_1, x_2 and x_3 are randomly assigned values. Then the value of logit is calculated by (6.4). Next, the obtained logit value is put in (6.2) and the probability of not observing a failure is calculated. Finally, the failure status variable z is determined by (6.3). A sample generated quality data with four runs is illustrated as the following:

x_1	x_2	x_3	z
1	0	1	0
1	1	1	1
0	0	0	0
0	1	0	0

Now let us analyze the generated data more. Assume all process variables x_1, x_2 and x_3 take a value of 1. Then, the conditional expected value of the logit (6.4) is $E(y|\vec{x})$ = 1.999 – 1 – 1 – 1 = −1.001 and by logistic function (6.2), we have $f(y)$ = 0.27 which ends up with a failure event or z = 1. Conversely, if all process variables take the 0 value, then the conditional expected value of the logit y is equal to the positive value +1.999 and by logistic function (6.2) we have $f(y)$ = 0.88 which ends up with a success status or z = 0. Obviously, there are totally 8 different combinations of effective process variables. The resulting expected logistic function values for all possible combinations of effective process variables are listed in Table 6.1.

Table 6.1 Expected Logistic function for all combinations of effective process variables

x_1	x_2	x_3	$E(y\|\vec{x})$	$f(y)$	$f'(y)$
0	0	0	1.999	0.88	0.12
1	0	0	0.999	0.73	0.27
0	1	0	0.999	0.73	0.27
0	0	1	0.999	0.73	0.27
1	1	0	– 0.001	0.49	0.50
1	0	1	– 0.001	0.49	0.50
0	1	1	– 0.001	0.49	0.50
1	1	1	– 1.001	0.27	0.73

In Table 6.1, $f(y)$ donates the probability of not observing a failure and $f'(y)$ donates the probability of observing a failure. Obviously, $f'(y) = 1 - f(y)$ is satisfied. Assume the logit (6.4) and related formulas are frequently used to generate a large quality data. In this case, different combinations of effective process variables regularly end up with failure or success events with the probabilities illustrated in Table 6.1. When the Apriori algorithm is applied on this generated data such that the consequent of rules is constrained to the failure /success events, then these regularities will be mined in the form of 16 association rules if their support and confidence are above the specified thresholds. The confidences of rules, with the success event as consequent, are consistent with the probabilities of not observing a failure event or $f(y)$ illustrated in Table 6.1. Similarly, the confidences of rules, with the failure event as consequent, are consistent with the probabilities of observing a failure event or $f'(y)$ illustrated in Table 6.1. The complete list of these 16 association rules is provided in Appendix A. Here, we assume that the data analyst is interested in just failure associations with a minimum confidence of 50%. Eight association rules out of 16 association rules listed in Appendix B include failure event as the consequent. However, only four of these failure association rules have a confidence satisfying the 50% threshold. Hence, these four rules are considered here as the *important failure* association rules and are listed in Figure 6.1 in the following.

$\{\{x_1 = 1\},\{x_2 = 1\},\{x_3 = 0\}\} \rightarrow \{z = 1\} < \text{confidence} \cong 50\% >$

$\{\{x_1 = 1\},\{x_2 = 0\},\{x_3 = 1\}\} \rightarrow \{z = 1\} < \text{confidence} \cong 50\% >$

$\{\{x_1 = 0\},\{x_2 = 1\},\{x_3 = 1\}\} \rightarrow \{z = 1\} < \text{confidence} \cong 50\% >$

$\{\{x_1 = 1\},\{x_2 = 1\},\{x_3 = 1\}\} \rightarrow \{z = 1\} < \text{confidence} \cong 73\% >$

Figure 6.1 Important failure association rules in sufficiently large generated data

These four associations in Figure 6.1 are meaningful and are not redundant as they show strong cause and effect rules between process variables and the failure events. Here, the only difference with reality is that we already know about these rules. Hence, if one GP method prunes some of the discovered important failure rules, it will result in information loss as some of the mined root causes of failure incidences in the system are removed and are not presented to data analyst. As a result, important failure rules in the Figure 6.1 can be used to measure the performance of rule reduction methods from the view of information loss. The more a GP method preserve mined important failure rules, the better it is in information preservation issue.

The data in reality would be complicated and consequently association rules mined from them would contain many trivial or redundant rules. To make our artificial quality data reflect this reality as well, we can add some subject-irrelevant input variables into the data. Here, we add two additional process variables x_4 and x_5 to our case, which have nothing to do with the failure /success status. We call them *ineffective* process variables. They are explained in the following.

The fourth ineffective process variable is x_4, which does not have any effect on failure /success event. Rather, it is completely dependent on the effective process variables. When at least two effective process variables are active, i.e. they take a value of 1, then x_4 is equal to 1. On the contrary, when at most one of the effective

process variables are active, then x_4 is equal to 0. This relation can be summarized as follows: if $x_1 + x_2 + x_3 \geq 2$ then $x_4 = 1$ otherwise $x_4 = 0$.

In real quality data, such failure-irrelevant relations are prevalent. One example relation could be , 'if the job is done by the machine *A* (equivalently $x_1 = 1$) and the used raw material is supplied by the company *B* ($x_2 = 1$) then the package *C* is to be used ($x_4 = 1$). When we mine failure rules from a dataset including the variable x_4, some redundant associations may turn out such as the following rule:

$$\{\{x_1 = 1\},\{x_2 = 1\},\{x_3 = 0\},\{x_4 = 1\}\} \rightarrow \{z = 1\}$$

The rule above is a more complex form of the first failure rule illustrated in Figure 6.2. The extra condition $\{x_4 = 1\}$ in the antecedent adds no more information and just makes the rule more complex. Hence, this rule is redundant and we expect rule reduction methods to prune such rules as much as possible.

The fifth ineffective process variable x_5 neither has any effect on failure/success event nor depends on some effective process variables. This variable just randomly gets some values but is included in quality data. In real quality data and any other types of data, there may be many data attributes that are part of data but do not have any association or correlation with others. They are just recorded and stored with other data attributes. Although they may be attributes that carry important information, they are irrelevant with the attributes under specific association study. Similarly, the variable x_5 is unrelated to failure event but it can be an important factor in other events. We consider the discrete uniform distribution with possible values of 1, 2, and 3 for the variable x_5. Clearly, any rule with an antecedent condition involving the process variable x_5 is redundant. The general definition for redundant rules is provided below;

> **Definition 6.1**: A rule is called a *redundant rule* if at least one of its conditions in the antecedent involves one of the ineffective process variables x_4 or x_5.

Finally, a sample generated dataset including all considered variables and with five runs would be as following:

x_1	x_2	x_3	x_4	x_5	z
1	1	0	1	1	0
1	0	1	1	3	1
0	0	0	0	1	0
0	1	0	0	3	0
0	1	1	1	2	0

If we have *N* runs, then we will have a dataset with *N* data rows. If we generate *m* datasets each one with *N* data rows, we will have *m* different datasets of D_1, D_2, …, D_m. Then, we can apply Apriori on each dataset and have *m* different rule sets of R_1, R_2, …, R_m. After that, we can apply different rule reduction method on obtained rule sets to see the performance of them. Finally, we can use some measures to evaluate the performance of applied GP methods on different rule and data sets. The whole performance evaluation process for GP methods is depicted in Figure 6.2.

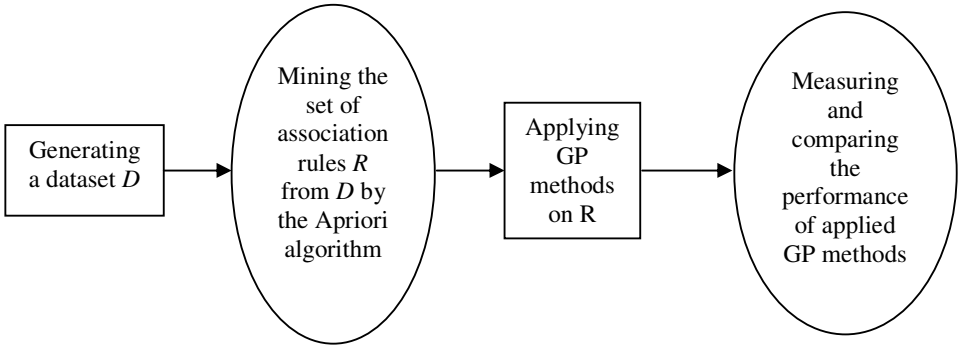


Figure 6.2 Performance evaluation process for GP methods

However, the performance evaluation process can also be done more efficiently by using experimental design. For example, we can consider two factors of *N* and *AT*. The factor *N* denotes the row size of datasets to be generated. If we would like to generate datasets with 100 and 10,000 rows, then *N* will have two levels as 100 and 10,000. The factor *AT* denotes the approach type (GP method) applied on rule sets. The number of levels of factor *AT* is equal to the number of rule reduction methods under comparison. If two GP methods of *A* and *B* are to be compared, then the factor *AT* will have two levels, first level indicating *A* method and second one indicating *B* method.

When the factors and their levels are determined, an experimental design can be selected. Assume the considered factors are *N* and *AT* each one with two levels as discussed earlier. The 2^2 full factorial design (See Table 6.2) can be selected for this case as it can accommodate two factors each with two levels.

Table 6.2 The used design for controlled experiment with two 2-level factors

N	AT	RN	FRN	SRN	RRN	RFRN
1	1					
1	2					
2	1					
2	2					

In the experimental design in Table 6.2, the first run indicates an experiment by which a dataset with 100 data rows will be generated. Then on the generated dataset *A* method will be applied. The second run indicates that *B* method will be applied on the same dataset already generated and used in the first run. The third run means a new dataset with 10,000 data rows will be generated and then *A* method will be applied on it. Finally, the fourth run indicates that *B* method will be applied on the same dataset generated and used in the third run. Hence, in the four runs above two

datasets, respectively with 100 and 10,000 rows, will be generated and then on each generated dataset both methods of *A* and *B* will be applied.

In Table 6.2, *RN* is the number of mined rules in each experiment. Apparently, the approach type *AT* will not affect *RN*. Also *FRN* is the number of mined failure rules between all mined rules. Clearly $FRN \leq RN$ is always satisfied. The characteristics *RN* and *FRN* are determined after Apriori is applied and the rules are mined and these are not affected by the GP methods. However, it is not true for the other characteristics, i.e. *SRN*, *RRN*, and *RFRN*. Here, *SRN* is the number of simplified rules, i.e. number of constructed clusters (or group of rules) and number of unclustered rules. *SRN* is determined when the grouping step is applied by GP methods. *RRN* is the number of remained rules after the pruning step. Clearly $RRN \leq RN$ is always satisfied. Finally, *RFRN* is the number of remained true rules after the pruning step. Obviously, $RFRN \leq FRN \leq RN$ is always satisfied. The summary of all characteristics and the relations between them are given in the following table.

Table 6.3 Summary of characteristics and their relations with each other

Characteristics:	**Relations:**
RN = # of mined rules after Apriori (after mining)	
FRN = # of failure rules after Apriori (after mining)	FRN ≤ RN
SRN = # of simplified rules after grouping step	SRN ≤ RRN ≤ RN
RRN = # of remained rules after pruning	RRN ≤ RN
RFRN = # of remained failure rules after pruning	RFRN ≤ FRN ≤ RN

Example: To clarify how different characteristics are calculated, consider the following example. Assume the set of all mined rules is $R = \{r_1, r_2, r_3, r_4, r_5\}$, where the rules r_1 and r_5 are two different failure rules from Figure 6.1 and the other rules are not failure rules. Here we have, $RN = 5$ and $FRN = 2$. Now assume we select the first GP method ($AT = 1$) to group and prune the mined rules. We apply the grouping step of first GP method on *R* and it results in the set of grouped rules $G = \{\{r_1, r_2\}, \{r_3, r_4\}\}$. In other words, the rules r_1 and r_2 are grouped together. The same thing

occurs for r_3 and r_4. But r_5 remains ungrouped. Here, $SRN = 3$ because we have 2 clusters of rules and 1 unclustered rule. Now assume we apply the pruning step of the first GP method and it results in removal of r_1 and r_4. Therefore, we have $RRN = 5 - 2 = 3$ as two rules are removed and $RFRN = 2 - 1 = 1$ as one of the failure rules (r_1) is removed. ■

So far, we explained how different datasets are collected and the rule reduction methods are applied on them. When all experiments are done we will have different calculated values for all characteristics in hand. Now let us measure the performance of the applied GP methods and to evaluate their strength from the different criteria. To this end, we have developed five evaluation measures each one trying to measure the performance of methods from a specific point of view, as listed in Table 6.4.

Table 6.4 Summary of the evaluation measures for GP methods

Measures	Range	Best case	Worst case
$M_1 = \frac{RFRN}{FRN}$	$0 \le M_1 \le 1$	1	0
$M_2 = \frac{SRN}{RN}$	$\frac{1}{RN} \le M_2 \le 1$	$\frac{1}{RN}$	1
$M_3 = \frac{\#\ of\ \ pruned\ \ redundant\ \ rules}{\#\ of\ \ redundant\ \ rules} = 1 - \frac{RRN - RFRN}{RN - FRN}$	$0 \le M_3 \le 1$	1	0
$M_4 = \frac{RFRN}{RRN}$	$0 \le M_4 \le 1$	1	0
$M_5 = \frac{w_1 M_1 + w_2 M_3}{w_1 + w_2}$	$0 \le M_5 \le 1$	1	0

The first measure (M_1) measures how much a GP method preserves failure rules. The best case is when all failure rules are preserved and not pruned i.e. $M_1 = 1$, and the

worst case is when all failure rules are pruned i.e. $M_1 = 0$. The second measure measures the smallness of clustered and summarized rules by a GP method. The best case is when all rules are grouped under one cluster i.e. $M_2 = 1 / RN$ and the worst case is when none of rules are grouped i.e. $M_2 = 1$. The third measure measures how much a GP method prunes redundant rules. The best case is when all redundant rules are pruned. In this case $M_3 = 1$. The worst case is when none of redundant rules are pruned. In this case $M_3 = 0$. The fourth measure M_4 tries to measure the ratio of remained failure rules over all remained rules. The best case is when all failure rules are kept and all redundant rules are removed. In this case $M_4 = 1$. The worst case is when all failure rules are removed. In this case $M_4 = 0$. However, this measure can be tricky. Consider the case in which just one failure rule is kept and the rest of failure and redundant rules are all removed. In such a case, this measure still gives us a value of 1 but this is not the best case as just one failure rule is preserved. The last measure M_5 is developed to solve this deficiency. The fifth measure is a combined one and simultaneously tries to measure the performance of a GP method from the two criteria; information preservation degree and pruning strength. The best case is when all failure rules are preserved and furthermore all redundant rules are pruned. In this case $M_5 = 1$. The worst case is when all failure rules are pruned and furthermore none of redundant rules are pruned. In this case $M_5 = 0$.

The next step is to run the designed experiments and collect data. When data are collected, the described measures in Table 6.4 can be calculated. Then the collected data can be analyzed to see which factors significantly affect which measures. Here, the analysis of variance and non-parametric tests can be useful. By the results of these tests we can evaluate the performance of rule reduction methods over different measures. To illustrate the whole process of performance evaluation, we conduct a controlled experiment in the following section to evaluate the performance of the metarules method and the proposed method on the generated data.

6.2 Application

We developed a new comparison basis for evaluation of different approaches in grouping and pruning discovered association rules. In Chapter 5, we showed that on some benchmark datasets our proposed approach groups and prunes the discovered rules better than the metarules method, especially when low thresholds are specified in the rule mining step. Now we will use the presented comparison basis in this chapter to analyze the grouping /pruning performance of both approaches more in the quality data domain.

The initial data are generated by the explained logistic regression model including the logit in equation (6.4), the logistic function in equation (6.2) and the failure status (6.3). The explained ineffective process variables x_4 and x_5 are used as well. Therefore, in our simulation model, we have five random variables. To vary the sequence of random numbers, we vary the initial seeds on which the sequence of random numbers is based. Hence, in generating each dataset, a different seed is used. When data are generated, the Apriori algorithm with minimum support of 1% and minimum confidence of 50% is applied. The antecedents of rules are confined to have at most four conditions. The consequent of rules is confined to the failure event or $z = 1$. The 2^2 full factorial design, which is illustrated in Table 6.2, is selected but this time four replicates are considered. The first level of *AT* is assumed to be the metarules method and the second level of *AT* is assumed to be the proposed approach. Briefly:

N = data row number (level 1 = 100; level 2 = 10,000)
AT = Approach Type (level 1 = the metarules method; level 2 = proposed method)

For any response variable (here characteristics in Table 6.3) totally 16 values are collected. Then the values of all five measures illustrated in Table 6.4 are calculated. For the 5th measure both weights w_1 and w_2 are set to 1. The collected characteristic values as well as calculated measures for all runs are summarized in Table 6.5 below.

Table 6.5 Collected characteristics and measures values

N	AT	RN	SRN	RRN	FRN	RFRN	M_1	M_2	M_3	M_4	M_5
1	1	97	52	62	3	3	1	0.536	0.366	0.048	0.683
1	2	97	42	52	3	3	1	0.433	0.473	0.058	0.737
1	1	78	49	54	2	2	1	0.628	0.297	0.037	0.649
1	2	78	33	43	2	2	1	0.423	0.446	0.047	0.723
1	1	79	39	47	3	3	1	0.494	0.413	0.064	0.707
1	2	79	29	38	3	3	1	0.367	0.533	0.079	0.767
1	1	88	52	59	3	3	1	0.591	0.333	0.051	0.667
1	2	88	33	44	3	3	1	0.375	0.512	0.068	0.756
2	1	71	42	47	2	2	1	0.592	0.328	0.043	0.664
2	2	71	37	42	2	2	1	0.521	0.403	0.048	0.702
2	1	96	53	62	4	4	1	0.552	0.370	0.065	0.685
2	2	96	41	50	4	4	1	0.427	0.500	0.080	0.750
2	1	82	46	53	3	3	1	0.561	0.359	0.057	0.680
2	2	82	39	46	3	3	1	0.476	0.449	0.065	0.725
2	1	76	47	52	2	2	1	0.618	0.306	0.039	0.653
2	2	76	37	42	2	2	1	0.487	0.444	0.048	0.722

In Table 6.5, the measure M_1 always is equal to 1 and its value never changes. This means that in the set of conducted experiments above, both GP methods (the metarules and the proposed approaches) preserve all mined important failure rules. Hence, none of the applied GP methods has any advantage over the other one from the aspect of information loss. We continue with other four measures of M_2, M_3, M_4 and M_5. Given the collected experiment data, we can apply analysis of variance or briefly ANOVA on the data in Table 6.5 to analyze the effect of factors *N* and *AT* on different measures and to find significant ones. Before going through that, we should check the normality assumption required by ANOVA. This analysis method requires the errors to be normally and independently distributed with mean zero and constant but unknown variance σ^2. To this end, analyses and standard charts for residuals, provided by Minitab software, are used to investigate normality assumption of the model. These plots are provided in Appendix (B).

The charts and analysis results in Appendix (B) imply that the normality assumption is roughly satisfied for the measures M_2, M_3, and M_5. As measures M_2, M_3, and M_5 seem to comply with normality assumption, we used just ANOVA test to analyze

them. However, the measure M_4 does not seem to conform to the normality assumption. To solve this, we used two data transformation techniques for measure M_4 and named the resulting measures TM_4 and LM_4. Then we investigated the residuals plots for them. The charts for TM_4 and LM_4 are also in the Appendix (B). The results for transformed data still do not assure the normality. Hence, for measure M_4 we applied ANOVA and further the non-parametric test of Kruskal-Wallis. We again used Minitab to run the tests. The results of all tests are provided in Appendix (B). According to the results, the approach type factor *AT* has a significant effect on measures M_2, M_3, and M_5. But data size factor *N* seems to be insignificant. In the next step, four measures of M_2, M_3, M_4 and M_5 versus the levels of factor *AT* are analyzed by one-way ANOVA analysis.

One-way ANOVA: M_2 versus AT

```
Source  DF       SS       MS      F      P
AT       1  0.07062  0.07062  29.06  0.000
Error   14  0.03402  0.00243
Total   15  0.10465

S = 0.04930   R-Sq = 67.49%   R-Sq(adj) = 65.16%

                           Individual 95% CIs For Mean Based on
                           Pooled StDev
Level  N     Mean    StDev  ---+---------+---------+---------+------
1      8  0.57150  0.04458                      (-----*-----)
2      8  0.43863  0.05360  (-----*-----)
                           ---+---------+---------+---------+------
                            0.420     0.480     0.540     0.600

Pooled StDev = 0.04930


Tukey 95% Simultaneous Confidence Intervals
All Pairwise Comparisons among Levels of AT

Individual confidence level = 95.00%

AT = 1 subtracted from:

AT     Lower    Center     Upper    -+---------+---------+---------+--
2   -0.18574  -0.13288  -0.08001    (--------*--------)
                                   -+---------+---------+---------+--
                                 -0.180    -0.120    -0.060     0.000
```

Figure 6.3 Results of one-way ANOVA on measure M_2

Figure 6.3 presents the output of the Minitab software for the measure M_2 versus two levels of the factor *AT*, i.e. the metarules approach and the proposed approach. The confidence intervals on each individual approach mean are provided and the means are compared using Tukey's method. Note that the Tukey method is presented using the confidence interval format. Apparently for this measure the Tukey confidence interval does not include zero and therefore we conclude that the means of approaches are different. Remember that the second measure measures the smallness of summarized rules and the lesser this measure is the better the approach groups rules. Here, the mean of proposed approach is less than the mean of the metarules approach. Hence, we conclude that our approach groups more rules.

One-way ANOVA: M_3 versus AT

```
Source  DF       SS       MS      F      P
AT       1  0.06101  0.06101  37.22  0.000
Error   14  0.02295  0.00164
Total   15  0.08396

S = 0.04049   R-Sq = 72.67%   R-Sq(adj) = 70.71%

                           Individual 95% CIs For Mean Based on
                           Pooled StDev
Level  N     Mean    StDev  -------+---------+---------+---------+--
1      8  0.34650  0.03803  (-----*-----)
2      8  0.47000  0.04280                          (-----*-----)
                           -------+---------+---------+---------+--
                                0.350     0.400     0.450     0.500

Pooled StDev = 0.04049


Tukey 95% Simultaneous Confidence Intervals
All Pairwise Comparisons among Levels of AT

Individual confidence level = 95.00%

AT = 1 subtracted from:

AT    Lower   Center    Upper  ----+---------+---------+---------+--
2   0.08008  0.12350  0.16692                  (-------*------)
                              ----+---------+---------+---------+--
                                  0.000     0.060     0.120     0.180
```

Figure 6.4 Results of one-way ANOVA on measure M_3

Now let's consider the third measure. Figure 6.4 presents the output from Minitab for the third measure M_3 for the metarules and the proposed approach. Clearly, the Tukey confidence interval also in this case does not include zero and therefore, we conclude that the means of approaches are different. As already explained, the third measure measures how much a GP method prunes redundant rules and the closer this measure is to 1 the more the approach prunes redundant rules. Here, the mean of proposed approach is closer to 1 than the mean of the metarules approach. Hence, we conclude that our method prunes the redundant rules more as well.

One-way ANOVA: M4 versus AT

```
Source  DF        SS        MS     F      P
AT       1  0.000495  0.000495  3.30  0.091
Error   14  0.002102  0.000150
Total   15  0.002597

S = 0.01225   R-Sq = 19.06%   R-Sq(adj) = 13.28%


                            Individual 95% CIs For Mean Based on
                            Pooled StDev
Level  N     Mean    StDev  --------+---------+---------+---------+-
1      8  0.05050  0.01077  (----------*----------)
2      8  0.06163  0.01357              (----------*----------)
                            --------+---------+---------+---------+-
                                  0.0480    0.0560    0.0640    0.0720

Pooled StDev = 0.01225


Tukey 95% Simultaneous Confidence Intervals
All Pairwise Comparisons among Levels of AT

Individual confidence level = 95.00%


AT = 1 subtracted from:

AT     Lower   Center    Upper  ---+---------+---------+---------+------
2   -0.00201  0.01113  0.02426            (------------*------------)
                               ---+---------+---------+---------+------
                               -0.010     0.000     0.010     0.020
```

Figure 6.5 Results of one-way ANOVA on measure M_4

The next measure is the M_4. The ANOVA and other analysis results for this measure are provided in Appendix (B). The results indicate that neither *N* nor *AT* have a significant effect on this measure. We also applied the one-way ANOVA test for this measure for the two GP methods and the results are provided in Figure 6.5. The factor *AT* seems to have a moderate effect on this measure. Its p-value is equal to 0.091 which is not far from the considered level of significance 0.05. The Tukey confidence interval in this case hardly includes zero and is not surrounding it. To analyze the effect of approach types on the measure M_4, the main effect and interaction plots are drawn and are available in Appendix (B). These charts also indicate that the mean of proposed approach for this measure is higher than the mean of the metarules method.

One-way ANOVA: M5 versus AT

```
Source  DF        SS        MS      F      P
AT       1  0.015252  0.015252  37.23  0.000
Error   14  0.005735  0.000410
Total   15  0.020988

S = 0.02024   R-Sq = 72.67%   R-Sq(adj) = 70.72%

                            Individual 95% CIs For Mean Based on
                            Pooled StDev
Level  N     Mean    StDev  -------+---------+---------+---------+--
1      8  0.67350  0.01905  (-----*------)
2      8  0.73525  0.02137                          (-----*-----)
                            -------+---------+---------+---------+--
                                 0.675     0.700     0.725     0.750

Pooled StDev = 0.02024


Tukey 95% Simultaneous Confidence Intervals
All Pairwise Comparisons among Levels of AT

Individual confidence level = 95.00%

AT = 1 subtracted from:

AT    Lower   Center    Upper  -------+---------+---------+---------+--
2   0.04004  0.06175  0.08346                       (-------*------)
                              -------+---------+---------+---------+--
                                   0.000     0.030     0.060     0.090
```

Figure 6.6 Results of one-way ANOVA on measure M_5

The last measure is the M_5. Figure 6.6 presents the output from Minitab for this measure. Here again, the Tukey confidence interval does not include zero and therefore we conclude that the means of methodes are different. The measure M_5 simultaneously measures success in information preservation and success in redundancy removal. The more a GP method preserves failure rules and prune redundant ones, the closer this measure is to 1. Here, the means of the proposed and the metarules approaches are 0.73525 and 0.67350, respectively. Therefore, the mean of the proposed method is closer to 1 than the mean of the metarules method. In this application, both methods keep all mined important failure rules. For this measure, the mean of our proposed method is larger because our method prunes more redundant rules.

In this chapter, a new basis is introduced to compare the performance of methods in grouping and pruning discovered association rules. Different measures are proposed to evaluate methods with respect to three criteria of information loss, grouping strength and pruning strength. The introduced basis is used to compare the proposed method in Chapter 4 with the metarules method. Based on the analysis results, three conclusions are achieved: both methods preserve the defined important information in the same level, the proposed method groups more rules, and the proposed method prunes more redundant rules. This method comparison basis is designed to be applicable in the quality data. However, it can be developed to be applicable in the classification data in general or in other types of categorical data. If continuous probability distributions are considered for input variables, then the application domain maybe extended to continuous data as well.

CHAPTER 7

CONCLUSION AND FUTURE WORK

A successful application of association rules require appropriate initial analysis and threshold settings for support and confidence in the rule mining process. In general, setting low thresholds is necessary to mine all interesting rules and to prevent information loss. However, the classic measures of support and confidence generate many redundant rules, especially when low thresholds are set for them. Hence, it is hard to interpret the huge number of mined rules.

Many approaches are introduced to reduce the set of mined association rules and make them more understandable. The approaches, based on "similarity" or "redundancy" among association rules, are very successful for rule reduction purposes. They can be categorized into two; grouping and pruning the association rules. Grouping techniques try to summarize the set of discovered rules by clustering the "similar" rules. Pruning techniques try to reduce the number of mined or to be mined association rules by detecting the "redundant" rules and removing them.

The focus of this thesis is on developing an effective rule reduction method particularly for applications requiring low support-confidence thresholds. There are some pre-developed methodologies that try to organize the large sets of mined association rules into human-tractable rule sets. However, many of these approaches may not be appropriate for applications requiring low threshold settings. Berrado and Runger [28] have developed the metarules method for organizing rules in sparse data. This method may encounter some problems in applications including low confident rules. We showed how this method can underestimate part of significant

overlaps which may cause effectiveness reduction in later grouping /pruning steps. In this study, the metarules method is improved to be more efficient in implementation and more effective in grouping and pruning rules particularly mined with low threshold settings. In our proposed method, the overlap and containment of rules are analyzed with new concepts and definitions. Then an algorithm is developed to mine overlaps /containments in a more efficient way. Experiments on some benchmarks datasets are conducted where the efficiency and effectiveness of proposed approach is compared with the metarules method. Results of experiments show that the proposed method scans the data less and also group and prune more rules.

Another important issue addressed in this thesis is how to evaluate the performance of different grouping and pruning methods. To this end, three general criteria are considered in evaluating and comparing the performance of such methods: information loss, grouping performance and pruning performance. A new performance comparison basis is introduced which enables data analysts to precisely evaluate the performance of different rule reduction methods. We used the presented basis to compare the performance of the metarules method with our method in the quality data. The results are consistent with the results of experiments on real benchmark datasets.

For the future work, we would like to combine the developed ODA algorithm with Apriori algorithm. Our intention is to have a modified Apriori algorithm, which can be reapplied on constructed NQ-set and that its computational complexity is improved. Another interesting research work would be to modify the Apriori algorithm as if it could mine association rules that are not overlapped by each other. In other words, we would like to embed grouping /pruning techniques inside initial rule mining steps such that the discovered association rules are already grouped and there is not any redundant rule between them. By doing that, the post-processing steps such as grouping and pruning discovered rules will be relaxed.

We developed a new basis which enables different rule reduction methods to be compared. In the future, we would like to conduct a larger series of experiments that

include variety of rule reduction methods and also many domain data. By doing this survey, we can find out which method is doing great on what kind of data and applications. Such information is very important and can save a lot of time for data analysts and help them to systematically and easily find the most suitable and effective method for their specific application.

REFERENCES

[1] Seref, O., and Xanthopoulos, P., 2009. Guest Editorial. *Journal of Combinatorial Optimization*, Volume 17, Number 1, p.1-2, January 2009.

[2] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P., 1996. From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 37-54

[3] Liu, B., Hsu, W., and Ma, Y., 1998. Integrating classification and association rule mining. *In Proceedings of the 4th international conference on knowledge discovery and data mining*, KDD, New York, USA, pp 80–86

[4] Agrawal, R., Imielinski, T., and Swami, A., 1993. Mining association rules between sets of items in large relational databases. *Proceedings of ACM SIGMOD international conference on management of data* 1993, 207-216.

[5] Shahbaz, M., Srinivas, S., Harding, J.A., and Turner, M., 2006. Product design and manufacturing process improvement using association rules. *Proc. IMechE*, Part B: J. Eng. Manufact., 220, pp. 243-254.

[6] Buddhakulsomsiri, J., Siradeghyan, Y., Zakarian, A., and Li, X., 2006. Association rule-generation algorithm for mining automotive warranty data, *International Journal of Production Research*, 44:14, 2749-2770

[7] Coenen, F. and Leng, P., 2007. The effect of threshold values on association rule based classification accuracy, *Data & Knowledge Engineering*, v.60 n.2, p.345-360, February, 2007

[8] Shaw, Gavin, Xu, Yue, and Geva, Shlomo, 2009. Interestingness Measures for Multi-Level Association Rules. *In Proceedings of ADCS 2009*, 30 Nov - 4 Dec, University of New South Wales, Sydney, Australia.

[9] Stephen E. Brossette, Alan P. Sprague, J. Michael Hardin, Ken B. Waites, Warren T. Jones, and Stephen A. Moser, 1998. Association Rules and Data Mining in Hospital Infection Control and Public Health Surveillance. *Journal*

of the American Medical Informatics Association, 5, (July/August), 373-381, 1998.

[10] Amir, A., Aumann, Y., Feldman, R., and Fresko, M., 2005. Maximal Association Rules: a Tool for Mining Associations in Text. *Journal of Intelligent Information Systems* (JIIS), 25(3), 333-345, November 2005.

[11] Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen H., and Inkeri Verkamo, A., 1994. Finding interesting rules from large sets of discovered association rules. *Proceedings of the third international conference on Information and knowledge management*, p.401-407, Gaithersburg, Maryland, United States

[12] Zaki, M.J., 2000. Generating non-redundant association rules. *In Proceedings of the Sixth ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY: ACM, 34-43.

[13] Brin, S., Motwani, R., and Silverstein, R., 1997(a). Beyond market basket: generalizing association rules to correlations. *SIGMOD-97*, 1997, 265-276.

[14] Brin, S., Motwani, R., Ullman, J., and Tsur, S., 1997(b). Dynamic Itemset Counting and Implication Rules for Market Basket Data. *In Ptoc. of the 1997 ACM-SIGMOD Int'l Conf on the Management of Data*, 255-264.

[15] Clark, P., and Boswell, P., 1991. Rule Induction with CN2: Some Recent Improvements. In Machine Learning. *Proc. of the Fifth European Conference*, 151-163

[16] Dhar, V., and Tuzhilin, A., 1993. Abstract-driven pattern discovery in databases. *IEEE Transactions on Knowledge and Data Engineering*, 5(6).

[17] Fukuda, T., Morimoto, Y., Morishita, S., and Tokuyama, T., 1996. Data Mining using Two-Dimensional Optimized Association Rules: Scheme, Algorithms, and Visualization. *In Proc. of the 1996 ACM-SIGMOD Int'l ConJ on the Management of Data*, 13-23.

[18] International Business Machines, 1996. IBM Intelligent Miner User's Guide, Version 1, Release 1.

[19] Liu, B., and Hsu, W., 1996. Post analysis of the learned rules. *In Proceedings of the 13th national conference on artificial intelligence*, Portland, OR, pp 828–834

[20] Liu, B., Hsu, W., and Chen, S., 1997. Using general impression to analyze discovered classification rules. *In Proceedings of the 3rd international conference on knowledge discovery and data mining*, Newport Beach, CA, USA, pp 31–36

[21] Morimoto, Y., Fukuda, T., Matsuzawa, H., Tokuyama, T., and Yoda, K., 1998. Algorithms for Mining Association Rules for Binary Segmentations of Huge Categorical Databases. *In Ptoc. of the 24th Very Large Data Bases Conf.*, 380-391

[22] Morishita, S., 1998. On Classification and Regression. *In Proc. of the First Int'l Conf on Discovery Science -- Lecture Notes in Artificial Intelligence* 1532: 40-57.

[23] Nakaya, A., and Morishita, S., 1999. Fast Parallel Search for Correlated Association Rules. *Unpublished manuscript*.

[24] Padmanabhan, B., and Tuzhilin, A., 1998. A belief-driven method for discovering unexpected patterns. *In Proceedings of the 4th international conference on knowledge discovery and data mining*, KDD, New York, USA, pp 94-100

[25] Silberschatz, A., and Tuzhilin, A., 1996. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Eng.*, 8(6):970-974, 1996.

[26] Webb, G.I., 1995. OPUS: an efficient admissible algorithm for unordered search, *Journal of Artificial Intelligence Research*, v.3 n.1, p.431-465

[27] Bayardo, R., Agrawal, R., and Gunopulos, D., 2000. Constraint-based rule mining in large, dense databases. *Data Min Knowl Discov* 4, 217-240.

[28] Berrado, A., and Runger, G.C., 2007. Using metarules to organize and group discovered association rules. *Data Mining and Knowledge Discovery* 14, 409-431.

[29] Chawla C, Davis, J Pandey G, 2004. On local pruning of association rules using directed hypergraphs. In: Proceedings of the 20th international conference on data engineering, Boston, MA, USA, p 832

[30] Gupta, K.G., Strehl, A., and Ghosh, J., 1999. Distance based clustering of association rules. *Proceedings of ANNIE, intelligent engineering systems through articial neural networks* 1999, 759-764.

[31] Lent, B., Swami, A., and Widom, J., 1997. Clustering association rules. *In Proceedings of the 13th international conference on data engineering, IEEE Computer Society*, Birmingham, UK, pp 220–231

[32] Ng, R., Lakshmanan, L.V.S., Han, J., and Pang, A., 1998. Exploratory mining and pruning optimizations of constrained associations rules. *In Proceedings of ACM SIGMOD international conference on management of data*, Seattle, WA, pp 13-24

[33] Srikant, R., Vu, Q., and Agrawal R., 1997. Mining association rules with item constraints. *In Proceedings of the 3rd international conference on knowledge discovery and data mining, KDD*, Newport Beach, CA, USA, pp 67-73

[34] Toivonen, H., Klemetinen, M., Ronkainen, P., Hatonen, K., and Mannila, H., 1995. Pruning and grouping discovered association rules. *Proceedings of the Mlnet workshop on statistics, machine learning, and discovery in databases*, 47-52

[35] Agrawal, R., and Srikant, R., 1994. Fast algorithms for mining association rules. *Proceedings of the 20th International Conference on Very Large Data Bases* 1994, 487-499.

[36] Han, J., Pei, J., and Yin, Y., 2000. Mining frequent patterns without candidate generation. *In Proceedings of ACM-SIGMOD International Conference on Management of Data*.

[37] Han, J., Pei, J., Yin, Y., and Mao, R, 2004. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Mining and Knowledge Discovery*, v.8 n.1, p.53-87

[38] Zaki, M.J., and Ogihara, M., 1998. Theoretical foundations of association rules. *In 3rd ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, June 1998.

[39] Zaki, M.J., and Hsiao, C.-J., 1999. CHARM: An efficient algorithm for closed association rule mining. *Technical Report 99-10*, Computer Science Dept., Rensselaer Polytechnic Institute.

[40] Pei, J., Han, J., and Mao, R., 2000. CLOSET: An efficient algorithm for mining frequent closed itemsets. *In Proceedings of ACM SIGMOD International Workshop on Data Mining and Knowledge Discovery*.

[41] Webb, G.I., 2000. Efficient search for association rules. *In Proceedings of the Sixth ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY: ACM, 99-107.

[42] Zheng, Z., Kohavi, R., Mason, L., 2001. Real world performance of association rule algorithms, *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, p.401-406, August 26-29, 2001, San Francisco, California

[43] Christian Hidber, 1999. Online association rule mining, *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, p.145-156, May 31-June 03, Philadelphia, Pennsylvania, United States

[44] Bayardo, R., Agrawal, R., 1999. Mining the most interesting rules, *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, p.145-154, August 15-18, 1999, San Diego, California, United States

[45] Weiß, C.H., 2008. Statistical Mining of Interesting Association Rules. *Statistics and Computing*. 18(2), pp. 185-194, 2008.

[46] Tan, P-N., Kumar, V., and Srivastava, J., 2002. Selecting the right interestingness measure for association patterns. *In Proceedings of the 8th ACM SIGKDD, international conference on knowledge discovery and data mining*, Madison, WI, USA, p 183

[47] Geng, L., and Hamilton, H.J., 2006. Interestingness Measures for Data Mining: A survey, *ACM Computing Surveys (CSUR)*, v.38, n.3, p.9-es, 2006

[48] Chmielewski, M. R., and Grzymala-Busse, J. W. 1996. Global discretization of continuous attributes as preprocessing for machine learning. *International Journal of Approximate Reasoning* 15 (1996), 319–331.

[49] Dougherty, J., Kohavi, R., Sahami, M., 1995. Supervised and unsupervised discretization of continuous features. *In Proceedings of the international conference on machine learning*, Tahoe City, CA, USA, pp 194–202

[50] Liu, H., Hussain, F., Tan, C.L., Dash, M., 2002. Discretization: an enabling technique. *Data Min Knowl Discov* 6(4):393-423

[51] María N. Moreno García, Isabel Ramos Román, Francisco J. García Peñalvo, Miguel Toro Bonilla, 2008. An association rule mining method for estimating the impact of project management policies on software quality, development time and effort. *Expert Systems with Applications: An International Journal*, v.34 n.1, p.522-529, January, 2008

[52] Stefan Born, Lars Schmidt-Thieme, 2004. Optimal Discretization of Quantitative Attributes for Association Rules, *In Classification, Clustering, and Data Mining Applications, Proceedings of the Meeting of the International Federation of Classification Societies* (IFCS), Illinois Institute of Technology, Chicago, 15--18 July 2004, pp. 287--296.

[53] Stephen D. Bay, 2001. Multivariate discretization of continuous variables for set mining, *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, p.315-319, August 20-23, 2000, Boston, Massachusetts, United States

[54] Yang, Y., Webb, G.I., 2002. A comparative study of discretization methods for naive-bayes classifiers. *In Proceedings of the Pacific Rim knowledge acquisition workshop*, PKAW, Tokyo, pp 159–173

[55] Yang, Y., G. I. Webb, and X. Wu, 2005. Chapter 6: Discretization Methods. In O. Maimon and L. Rokach (Eds.), *The Data Mining and Knowledge Discovery Handbook*. Berlin: Springer, pages 113-130.

[56] Srikant, R., and Agrawal, R., 1996. Mining quantitative association rules in large relational tables, *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, p.1-12, June 04-06, 1996, Montreal, Quebec, Canada

[57] Aumann, Y., Lindell, Y., 2003. A statistical theory for quantitative association rules. *J Intell Inf Syst* 20(3):255–283

[58] Friedman, J., and Fisher, N., 1999. Bump hunting in high-dimensional data. *Statistics and Computing*, Volume 9, Issue 2, Pages 123-143

[59] Asuncion, A., and Newman, D.J., 2007. UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science.

[60] Borgelt, C., and Kruse, R., 2002. Induction of Association Rules: Apriori Implementation. *15th Conference on Computational Statistics*

[61] Borgelt, C., http://www.borgelt.net//software.html, last updated on 28/05/2010

[62] SPSS Inc., 2007, Clementine 11.1, *Application Guide*

[63] SPSS Inc., 2007, Clementine 11.1, *Clementine Algorithms Guide*

[64] SPSS Inc., 2007, Clementine 11.1, *Node Reference*

[65] McCullagh, P., 1980. Regression models for ordinal data (with discussion). *Journal of the Royal Statistical Society*. Series B, Vol. 42, Issue 2, 109-142.

APPENDIX A

ASSOCIATION RULES IN THE GENERATED QUALITY DATA

$$\{\{x_1 = 0\},\{x_2 = 0\},\{x_3 = 0\}\} \rightarrow \{z = 0\} < \text{confidence} \cong 88\% >$$
$$\{\{x_1 = 0\},\{x_2 = 0\},\{x_3 = 0\}\} \rightarrow \{z = 1\} < \text{confidence} \cong 12\% >$$

$$\{\{x_1 = 1\},\{x_2 = 0\},\{x_3 = 0\}\} \rightarrow \{z = 0\} < \text{confidence} \cong 73\% >$$
$$\{\{x_1 = 1\},\{x_2 = 0\},\{x_3 = 0\}\} \rightarrow \{z = 1\} < \text{confidence} \cong 27\% >$$

$$\{\{x_1 = 0\},\{x_2 = 1\},\{x_3 = 0\}\} \rightarrow \{z = 0\} < \text{confidence} \cong 73\% >$$
$$\{\{x_1 = 0\},\{x_2 = 1\},\{x_3 = 0\}\} \rightarrow \{z = 1\} < \text{confidence} \cong 27\% >$$

$$\{\{x_1 = 0\},\{x_2 = 0\},\{x_3 = 1\}\} \rightarrow \{z = 0\} < \text{confidence} \cong 73\% >$$
$$\{\{x_1 = 0\},\{x_2 = 0\},\{x_3 = 1\}\} \rightarrow \{z = 1\} < \text{confidence} \cong 27\% >$$

$$\{\{x_1 = 1\},\{x_2 = 1\},\{x_3 = 0\}\} \rightarrow \{z = 0\} < \text{confidence} \cong 50\% >$$
$$\{\{x_1 = 1\},\{x_2 = 1\},\{x_3 = 0\}\} \rightarrow \{z = 1\} < \text{confidence} \cong 50\% >$$

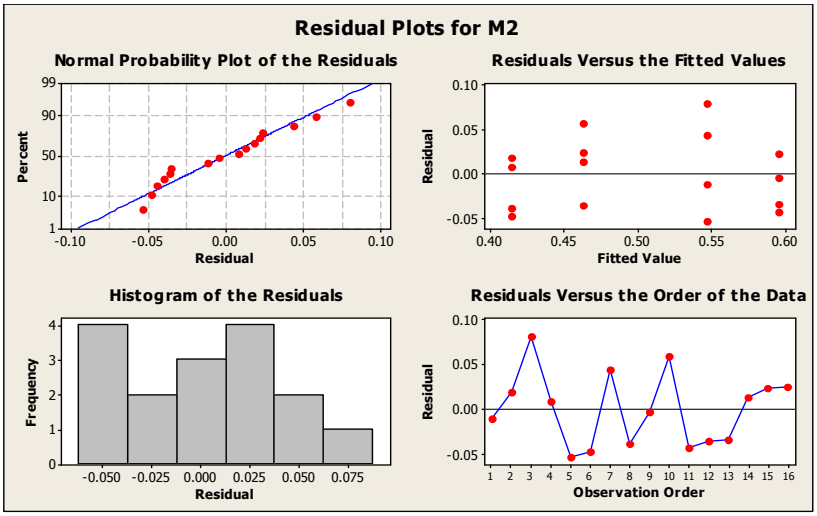
$$\{\{x_1 = 1\},\{x_2 = 0\},\{x_3 = 1\}\} \rightarrow \{z = 0\} < \text{confidence} \cong 50\% >$$
$$\{\{x_1 = 1\},\{x_2 = 0\},\{x_3 = 1\}\} \rightarrow \{z = 1\} < \text{confidence} \cong 50\% >$$

$$\{\{x_1 = 0\},\{x_2 = 1\},\{x_3 = 1\}\} \rightarrow \{z = 0\} < \text{confidence} \cong 50\% >$$
$$\{\{x_1 = 0\},\{x_2 = 1\},\{x_3 = 1\}\} \rightarrow \{z = 1\} < \text{confidence} \cong 50\% >$$

$$\{\{x_1 = 1\},\{x_2 = 1\},\{x_3 = 1\}\} \rightarrow \{z = 0\} < \text{confidence} \cong 27\% >$$
$$\{\{x_1 = 1\},\{x_2 = 1\},\{x_3 = 1\}\} \rightarrow \{z = 1\} < \text{confidence} \cong 73\% >$$

APPENDIX B

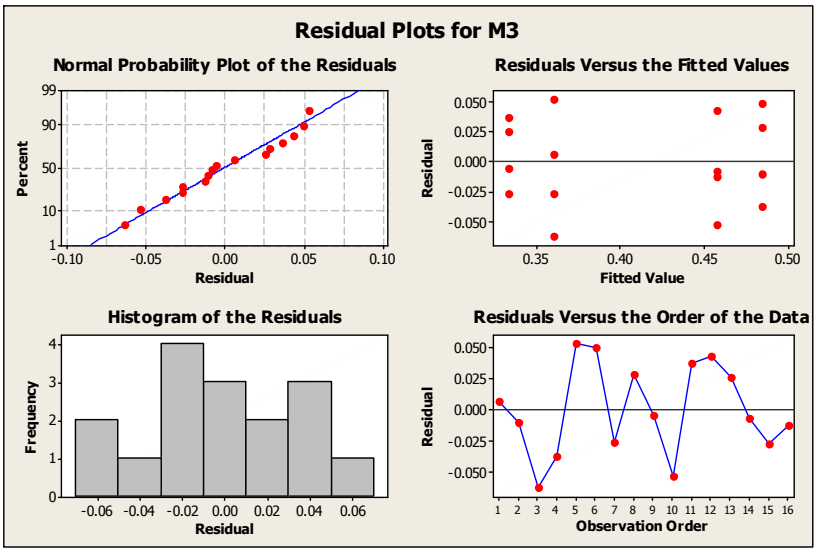
CHARTS & ANALYSIS RESULTS FOR EVALUATION MEASURES



General Linear Model: M2 versus N, AT

```
Factor  Type   Levels  Values
N       fixed       2  1, 2
AT      fixed       2  1, 2

Analysis of Variance for M2, using Adjusted SS for Tests
Source  DF    Seq SS    Adj SS    Adj MS      F      P
N        1  0.009361  0.009361  0.009361   4.93  0.045
AT       1  0.070623  0.070623  0.070623  37.23  0.000
Error   13  0.024663  0.024663  0.001897
Total   15  0.104647

S = 0.0435566   R-Sq = 76.43%   R-Sq(adj) = 72.81%

Unusual Observations for M2
Obs        M2       Fit    SE Fit  Residual  St Resid
  3  0.628000  0.547313  0.018861  0.080688      2.06 R

R denotes an observation with a large standardized residual.
```

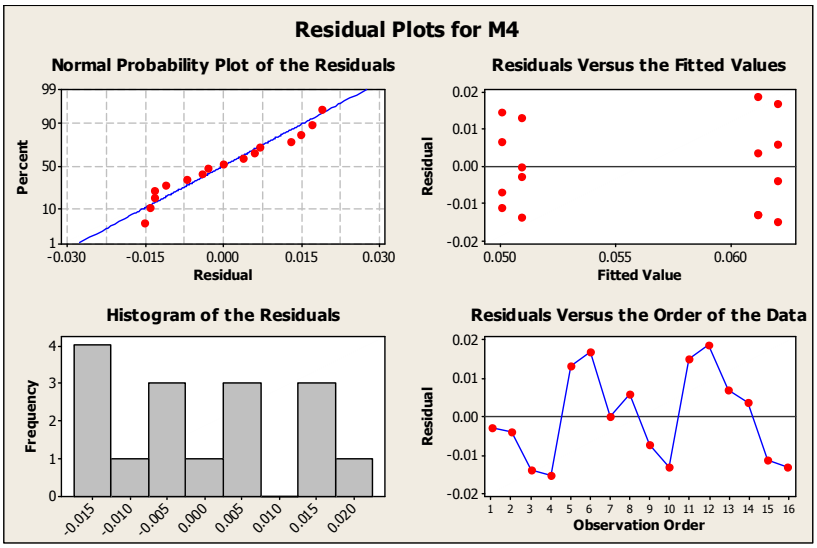
Figure B.1 Charts & analysis results for measure M_2



General Linear Model: M3 versus N, AT

```
Factor  Type   Levels  Values
N       fixed       2  1, 2
AT      fixed       2  1, 2


Analysis of Variance for M3, using Adjusted SS for Tests

Source  DF    Seq SS    Adj SS    Adj MS      F      P
N        1  0.002862  0.002862  0.002862   1.85  0.197
AT       1  0.061009  0.061009  0.061009  39.48  0.000
Error   13  0.020088  0.020088  0.001545
Total   15  0.083959


S = 0.0393092   R-Sq = 76.07%   R-Sq(adj) = 72.39%
```

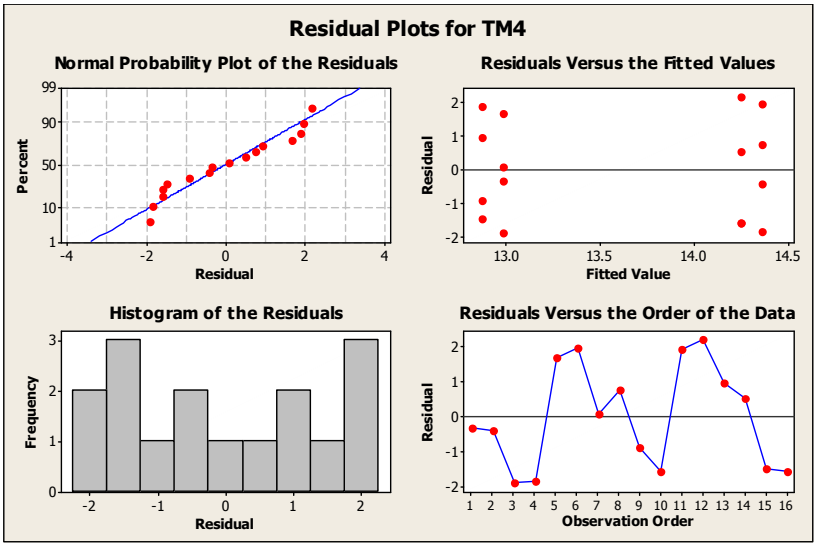
Figure B.2 Charts & analysis results for measure M_3



General Linear Model: M4 versus N, AT

```
Factor  Type   Levels  Values
N       fixed       2  1, 2
AT      fixed       2  1, 2


Analysis of Variance for M4, using Adjusted SS for Tests

Source  DF     Seq SS     Adj SS     Adj MS     F      P
N        1  0.0000031  0.0000031  0.0000031  0.02  0.893
AT       1  0.0004951  0.0004951  0.0004951  3.07  0.103
Error   13  0.0020988  0.0020988  0.0001614
Total   15  0.0025969


S = 0.0127062   R-Sq = 19.18%   R-Sq(adj) = 6.75%
```

Figure B.3 Charts & analysis results for measure M_4

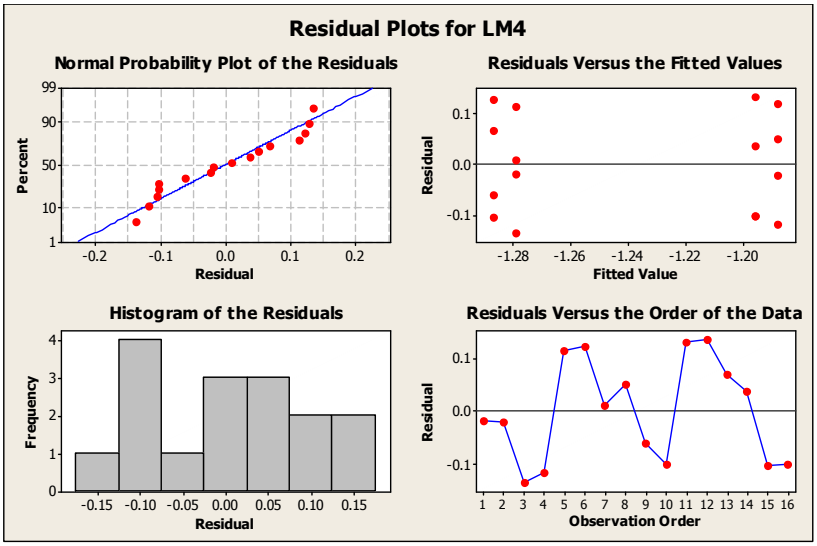


General Linear Model: TM4 versus N, AT

```
Factor  Type   Levels  Values
N       fixed       2  1, 2
AT      fixed       2  1, 2


Analysis of Variance for TM4, using Adjusted SS for Tests

Source  DF  Seq SS  Adj SS  Adj MS     F      P
N        1   0.050   0.050   0.050  0.02  0.889
AT       1   7.584   7.584   7.584  3.07  0.103
Error   13  32.090  32.090   2.468
Total   15  39.724


S = 1.57113   R-Sq = 19.22%   R-Sq(adj) = 6.79%
```

$$TM_4 = Arc\sin(\sqrt{M_4})$$

Figure B.4 Charts & analysis results for measure TM_4



General Linear Model: LM4 versus N, AT

```
Factor  Type   Levels  Values
N       fixed       2  1, 2
AT      fixed       2  1, 2


Analysis of Variance for LM4, using Adjusted SS for Tests

Source  DF   Seq SS   Adj SS   Adj MS     F      P
N        1  0.00023  0.00023  0.00023  0.02  0.887
AT       1  0.03319  0.03319  0.03319  3.08  0.103
Error   13  0.14029  0.14029  0.01079
Total   15  0.17371


S = 0.103882   R-Sq = 19.24%   R-Sq(adj) = 6.81%
```

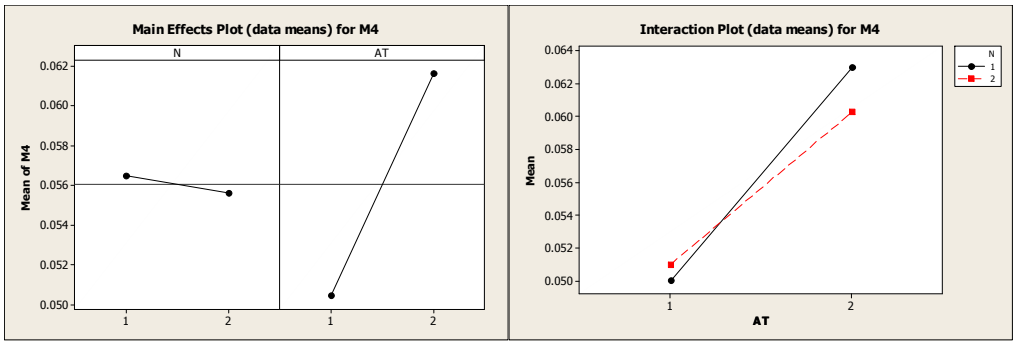
$$LM_4 = Log\left(\frac{M_4}{1-M_4}\right)$$

Figure B.5 Charts & analysis results for measure LM_4

Kruskal-Wallis Test: M4 versus AT

```
Kruskal-Wallis Test on M4

AT        N   Median  Ave Rank      Z
1         8  0.04950       6.6  -1.63
2         8  0.06150      10.4   1.63
Overall  16                8.5

H = 2.65  DF = 1  P = 0.104
H = 2.67  DF = 1  P = 0.102  (adjusted for ties)
```

Figure B.6 Results of Kruskal-Wallis Test on measure M_4



Factorial Fit: M4 versus N, AT

Estimated Effects and Coefficients for M4 (coded units)

Term	Effect	Coef	SE Coef	T	P
Constant		0.056063	0.003295	17.01	0.000
N	-0.000875	-0.000437	0.003295	-0.13	0.897
AT	0.011125	0.005563	0.003295	1.69	0.117
N*AT	-0.001875	-0.000937	0.003295	-0.28	0.781

S = 0.0131806 R-Sq = 19.72% R-Sq(adj) = 0.00%

Analysis of Variance for M4 (coded units)

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Main Effects	2	0.00049813	0.00049813	0.00024906	1.43	0.277
2-Way Interactions	1	0.00001406	0.00001406	0.00001406	0.08	0.781
Residual Error	12	0.00208475	0.00208475	0.00017373		
Pure Error	12	0.00208475	0.00208475	0.00017373		
Total	15	0.00259694				

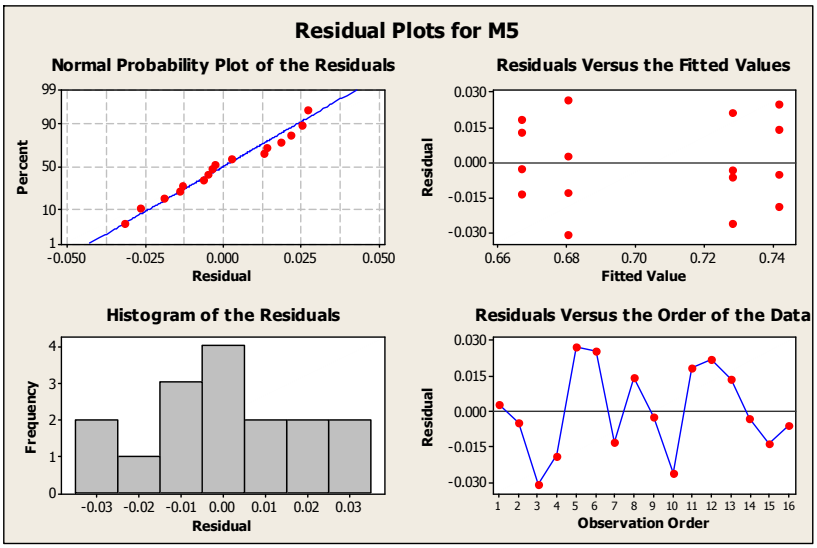
Estimated Coefficients for M4 using data in uncoded units

Term	Coef
Constant	0.0322500
N	0.0047500
AT	0.0167500
N*AT	-0.0037500

Alias Structure

I
N
AT
N*AT

Figure B.7 Main effects and ANOVA results for measure M_4



General Linear Model: M5 versus N, AT

```
Factor  Type   Levels  Values
N       fixed       2  1, 2
AT      fixed       2  1, 2


Analysis of Variance for M5, using Adjusted SS for Tests

Source  DF     Seq SS     Adj SS     Adj MS       F      P
N        1  0.0007290  0.0007290  0.0007290    1.89  0.192
AT       1  0.0152522  0.0152522  0.0152522   39.60  0.000
Error   13  0.0050065  0.0050065  0.0003851
Total   15  0.0209877


S = 0.0196244   R-Sq = 76.15%   R-Sq(adj) = 72.48%
```

Figure B.8 Charts & analysis results for measure M_5