

The copyright © of this thesis belongs to its rightful author and/or other copyright owner. Copies can be accessed and downloaded for non-commercial or learning purposes without any charge and permission. The thesis cannot be reproduced or quoted as a whole without the permission from its rightful owner. No alteration or changes in format is allowed without permission from its rightful owner.



**A MODIFIED MULTI-CLASS ASSOCIATION RULE FOR TEXT
MINING**



MOHAMMAD HAYEL AL REFAI

UUM

Universiti Utara Malaysia

**DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2015**

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from the from University Utara Malaysia, I agree that the University Library may make it freely available for inspection. I further agree that permission for copying of this project in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor or in his absence by the Assistant of Vice Chancellor of College of Arts and Sciences. It is understood that any copying or publication or use of this project or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to University Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to



Dean of Awang Had Salleh Graduate School of Arts and Sciences

UUM College of Arts and Sciences

Universiti Utara Malaysia

06010 UUM Sintok

Abstrak

Klasifikasi dan perlombongan peraturan berkait adalah tugas yang signifikan dalam perlombongan data. Integrasi antara penemuan peraturan berkait dengan klasifikasi dalam perlombongan data menghasilkan klasifikasi perkaitan. Salah satu kekurangan Pengklasifikasi Perkaitan adalah penghasilan bilangan peraturan yang besar bagi mencapai kejituan klasifikasi yang tinggi. Kajian ini memperkenalkan *Modified Multi-class Association Rule Mining (mMCAR)* yang mengandungi tiga prosidur; penemuan peraturan, cantasan peraturan dan pengumpulan kelas berdasarkan kumpulan. Prosidur penghasilan peraturan dan cantasan peraturan direkabentuk untuk mengurangkan bilangan peraturan klasifikasi. Manakala, prosidur pengumpulan kelas berdasarkan kumpulan menyumbang kepada peningkatan kejituanklasifikasi. Eksperimen ke atas koleksi data teks berstruktur dan tidak berstruktur yang diperolehi dari repositori UCI dan Reuters dilaksanakan untuk menilai Pengklasifikasi Perkaitan yang dicadangkan. Pengklasifikasi *mMCAR* yang dicadangkan telah ditanda aras dengan pengklasifikasi tradisional dan Pengklasifikasi Perkaitan sedia ada. Keputusan eksperimen menunjukkan bahawa Pengklasifikasi Perkaitan yang dicadangkan menghasilkan kejituan klasifikasi yang tinggi dengan menggunakan bilangan peraturan yang lebih kecil. Bagi koleksi data berstruktur, pengklasifikasi *mMCAR* telah menghasikan nilai purata 84.24% kejituan berbanding dengan *MCAR* yang memperolehi 84.23%. Walaupun perbezaan kejituan klasifikasi adalah kecil, pengklasifikasi *mMCAR* hanya menggunakan 50 peraturan manakala kaedah penanda aras melibatkan 60 peraturan. Dalam pada itu, *mMCAR* didapati setanding dengan *MCAR* apabila koleksi data tidak berstruktur digunakan. Kedua-dua pengklasifikasi menghasilkan 89% kejituan tetapi *mMCAR* menggunakan bilangan peraturan yang lebih kecil untuk membuat klasifikasi. Kajian ini menyumbang kepada domain perlombongan teks kerana klasifikasi automatik bagi data yang besar dan teragih boleh membantu proses perwakilan dan capaian teks.

Kata Kunci: Perlombongan data, Perlombongan teks, Klasifikasi, Klasifikasi perkaitan.

Abstract

Classification and association rule mining are significant tasks in data mining. Integrating association rule discovery and classification in data mining brings us an approach known as the associative classification. One common shortcoming of existing Association Classifiers is the huge number of rules produced in order to obtain high classification accuracy. This study proposes a Modified Multi-class Association Rule Mining (*mMCAR*) that consists of three procedures; rule discovery, rule pruning and group-based class assignment. The rule discovery and rule pruning procedures are designed to reduce the number of classification rules. On the other hand, the group-based class assignment procedure contributes in improving the classification accuracy. Experiments on the structured and unstructured text datasets obtained from the UCI and Reuters repositories are performed in order to evaluate the proposed Association Classifier. The proposed *mMCAR* classifier is benchmarked against the traditional classifiers and existing Association Classifiers. Experimental results indicate that the proposed Association Classifier, *mMCAR*, produced high accuracy with a smaller number of classification rules. For the structured dataset, the *mMCAR* produces an average of 84.24% accuracy as compared to *MCAR* that obtains 84.23%. Even though the classification accuracy difference is small, the proposed *mMCAR* uses only 50 rules for the classification while its benchmark method involves 60 rules. On the other hand, *mMCAR* is at par with *MCAR* when unstructured dataset is utilized. Both classifiers produce 89% accuracy but *mMCAR* uses less number of rules for the classification. This study contributes to the text mining domain as automatic classification of huge and widely distributed textual data could facilitate the text representation and retrieval processes.

Keywords: Data mining, Text mining, Classification, Associative classification.

Acknowledgement

First of all, I would like to thank god, whose grace has led me to this important moment of my life.

I would like to take this opportunity to express my thanks to my supervisor Dr. Yuhanis binti Yusof for her encouragement, support and guidance throughout this research project.

I would also dedicate this research to the spirit of my father, and a special thanks to my mother and brother for their ongoing support and advice. Last but not least, I would like to thank my wife and children who has been a source of inspiration and much support to me.

Finally, many thanks to my friends for supporting me.



List of Acronyms

AC	Associative Classification
ACCF	Association Classification based on Closed Frequent Itemsets
ACCR	Association Classification based on Compactness of Rules
ACN	Association Classifier with Negative Rules
BCAR	Boosting Association Rules
CACA	Class Based Association Classification
CAR	Class Association Rule
CBA	Classification based on Association Rule
CMAR	Classification based on Multiple Class-Association Rules
CPAR	Classification based on Predictive Association Rules
IG	Information Gain
JCSCP	Joint Confidence Support Class Prediction Method
MCAR	Multi-class Classification based on Association Rule
MMAC	Multi-class, Multi-label Associative Classification
mMCAR	Modified Multi-class Classification based on Association Rule
PRM	Pruning Method Partly Rule Match
RIPPER	Repeated Incremental Pruning to Produce Error Reduction
SVM	Support Vector Machine
TC	text classification
WEKA	Waikato Environment for Knowledge Analysis

Table of Contents

Permission to Use.....	ii
Abstrak	iii
Abstract	iv
Acknowledgement.....	v
List of Acronyms	vi
Table of Contents	vii
List of Tables.....	xii
List of Figures	xiv
Dissemination.....	xv
CHAPTER ONE	1
INTRODUCTION	1
1.1 Background	1
1.2 Research Motivation	6
1.3 Research Problem.....	6
1.4 Research Questions	8
1.5 Research Objectives	8
1.6 Research Scope	9
1.7 Research Contributions	9
1.8 Thesis Organization	10
CHAPTER TWO	11
RELATED WORK	11
2.1 Introduction	11
2.2 Data pre-processing.....	11
2.2.1 A Bag of Word Representation	11
2.2.2 A Numerical Vector Representation	12
2.2.2.1 Term Weighting	13

2.2.2.2	Feature Selection and Dimensionality Reduction	15
2.2.2.3	Document Frequency	17
2.3	Text Classification	17
2.3.1	K-nearest Neighbor	17
2.3.2	Naive Bayesian.....	19
2.3.3	Decision Trees.....	19
2.3.4	Support Vector Machine	20
2.4	Evaluation Measures in Text Classification.....	21
2.5	Association Classification.....	24
2.5.1	Classification Based on Association Rule.....	27
2.5.2	Multi-class Classification based on Association rule.....	28
2.5.3	Classification based on Multiple Association Rules.....	30
2.5.4	Classification based on Predictive Association Rule.....	31
2.5.5	Multi-class, Multi-label Association Classification Approach	32
2.5.6	Two-Phase Based Classifier Building.....	33
2.5.7	Class Based Association Classification	33
2.5.8	Association Classifier with Negative Rules.....	34
2.5.9	Association Classification based on Closed Frequent Itemsets	35
2.5.10	Boosting Association Rules	35
2.5.11	Association Classification based on Compactness of Rules	35
2.6	Rule Discover and Production in Association Classification	36
2.6.1	Apriori.....	37
2.6.2	Frequent Pattern Growth.....	39
2.6.3	Tid-list Intersection	40
2.7	Pruning Methods in AC	41
2.7.1	Database Coverage.....	43
2.7.2	Lazy Methods.....	44
2.7.3	Long Rules Pruning	46

2.7.4 Mathematical Based Pruning	47
2.7.4.1 Chi-Square Testing.....	47
2.7.4.2 Pessimistic Error Estimation	49
2.7.4.3 Pearson’s Correlation Coefficient Testing	50
2.7.5 Laplace Accuracy.....	51
2.7.6 Redundant Rule Pruning	52
2.7.7 Conflicting Rules	52
2.7.8 Compact Rule Set.....	53
2.7.9 I-Prune.....	54
2.7.10 PCBA-based Pruning	55
2.8 The Methods of Prediction.....	55
2.8.1 Single Rule Class Assignment	56
2.8.2 Class Assignment Based on Group of Rules	57
2.8.2.1 Weighted Chi-Square Method.....	57
2.8.2.2 Laplace based Method.....	58
2.8.2.3 Dominant Class and Highest Confidence Method	59
2.8.3 Predictive Confidence	60
2.9 Comparison of Association Classification Algorithm	61
2.10 Chapter Summary.....	68
CHAPTER THREE.....	69
RESEARCH METHODOLOGY.....	69
3.1 Introduction.....	69
3.2 Data collection	71
3.3 Data Pre-processing	73
3.3.1 Tokenisation.....	74
3.3.2 Stopwords Removal	74
3.3.3 Stemming	74
3.3.4 Data Representation	75
3.3.5 Feature selection.....	77

3.4 Design Classifier Model.....	77
3.4.1 Rule discovery.....	77
3.4.2 Rule ranking.....	79
3.4.3 Rule Pruning.....	79
3.4.4 Predicting of Test Data.....	80
3.5 Development classifier and Evaluation.....	81
3.6 Summary.....	82
CHAPTER FOUR.....	83
MODIFIED MULTICLASS ASSOCIATION RULE CLASSIFIER.....	83
4.1 Introduction.....	83
4.2 Proposed Classifier.....	83
4.3 CARs Discovery and Production.....	85
4.4 Rule Ranking.....	92
4.5 Pruning Method Partly Rule Match.....	93
4.6 Joint Confidence Support Class Prediction Method.....	97
4.7 Summary.....	101
CHAPTER FIVE.....	102
RESULTS AND DISCUSSION.....	102
5.1 Introduction.....	102
5.2 Rules Obtained Using Both the MCAR and mMCAR.....	103
5.3 Structured Data Set.....	106
5.3.1 Prediction Accuracy for UCI Data Set.....	106
5.3.2 Number of Rules for UCI Data Set.....	111
5.3.3 Win-Loss-Tie Record for UCI Data Set.....	114
5.3.4 Compression Variation Between AC Algorithms.....	115
5.3.5 Computational Time for UCI Data Set.....	116
5.4 Unstructured Dataset.....	117
5.4.1 Prediction Accuracy for Reuters Data Set.....	118
5.4.2 Number of rule for Reuters Data Set.....	120
5.4.3 The Win-Loss-Tie Record.....	123
5.4.4 Compression Variation between AC algorithms.....	124

5.4.5 Training and Testing Time for Reuter's Data Set.....	127
5.5 Summary	128
CHAPTER SIX	129
CONCLUSION AND FUTURE WORK	129
6.1 Conclusion	129
6.2 Rule Discovery Algorithm that Reduces Computational Time	129
6.3 Rule Pruning Algorithm that Reduces the Number of Classification Rules.....	130
6.4 Rule Prediction Algorithm that Improves Accuracy.....	130
6.5 Future Work	131
6.5.1 Multi-label in Text Classification	131
6.5.2 Discretisation.....	132
6.5.3 Pre-Pruning	132
6.6 Summary	133
REFERENCE.....	134
APPENDIX A.....	142
APPENDIX B.....	149
APPENDIX C.....	150
APPENDIX D.....	151



List of Tables

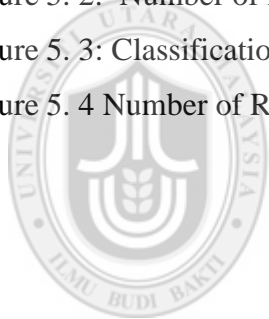
Table 2.1: Documents possible sets based on a query in IR	22
Table 2.2: Training Data	25
Table 2.3: Summary of AC algorithms	67
Table 3.1: Description of UCI Data Sets	72
Table 3.2: Number of documents (REUTERS-21578).....	73
Table 3.3: Examples of Item Found on Each Line.....	75
Table 3.4: Representation of Item.....	76
Table 4.1: Training data set.....	88
Table 4.2: Frequent Items	89
Table 4.3: Example data from Weather Dataset	89
Table 4.4: Candidate 1-ruleitemYES	90
Table 4.5: Candidate 1-ruleitem NO.....	90
Table 4.6: Candidate 2-ruleitemclass YES	91
Table 4.7: Candidate 2-ruleitemclass No.....	91
Table 4.8: Frequent items.....	92
Table 4.9: Rule Ranking	93
Table 4.10: Rule Pruning Using Weather Dataset	95
Table 4.11: Frequent item and rule ranking for Weather Dataset.....	95
Table 4.12: A Rule-Based Model.....	100
Table 4.13: Testing case.....	100
Table 4.14: Applicable Rules for Ts	101
Table 5. 1 Training Data Set	103
Table 5. 2 Ranked candidate rules produced by MCAR and mMCAR.....	104
Table 5. 3 Sample of rules by mMCAR and MCAR on UCI “Cleve” data set	105
Table 5. 4 Sample of rules by mMCAR and MCAR on Reuter’s “acq” data set ...	106
Table 5. 5 The Prediction Clasification Accuracy on UCI Data Sets	108
Table 5. 6 The Classification Accuracy Between mMCAR and all Algorithms	110
Table 5. 7 The Number of Rules for the UCI Data Sets	112
Table 5. 8 The Number of Rules Between mMCAR and all Algorithms	113
Table 5. 9 Won Loss-Tie Accuracy for UCI Dataset.....	114
Table 5. 10 Won-Loss-Tie Number of Rules for UCI Dataset	114
Table 5. 11 The variation of UCI data set between AC algorithms.....	115

Table 5. 12 Training Time for UCI Data Sets Using AC Algorithm.....	117
Table 5.13 Classification Accuracy on seven most populated Reuters data sets....	119
Table 5. 14 Classification Accuracy Between mMCAR and all Algorithms	120
Table 5. 15 Number of Rules using different pruning approach.....	121
Table 5. 16 Number of Rules Between mMCAR and all Algorithms	122
Table 5. 17 Results on Win/Lose/Tie for accuracy.....	123
Table 5. 18 Results on Win/Lose/Tie for number of rule	123
Table 5. 19 The variation of Reuter’s data set between AC algorithms	124
Table 5. 20 Training and Testing Time for Reuter’s Data Sets	127



List of Figures

Figure 2.1: Candidate Generation Apriori Algorithm.....	38
Figure 2.2: Database Coverage Pruning Methods.....	44
Figure 2.3: Single Rule Class Assignment Methods.....	56
Figure 2.4: Theoretical Framework.....	66
Figure 3.1: Research Methodology	70
Figure 3.2: Pre-processing Operation in Text Mining	73
Figure 4.1: mMCAR Steps.....	84
Figure 4.2: The mMCAR Algorithm	85
Figure 4.3: Production of Rule.....	87
Figure 4.4: Partly Rule Match Pruning Method.....	96
Figure 4.5: Joint Confidence Support Class Prediction	99
Figure 5. 1: Prediction Accuracy on UCI Data sets	109
Figure 5. 2: Number of Rules of the considered algorithms on UCI data sets	112
Figure 5. 3: Classification Accuracy of Reuters Data Sets	119
Figure 5. 4 Number of Rules Using Different Pruning Approaches.....	122



UUM
Universiti Utara Malaysia

Dissemination

Most part of this thesis has been published in international scientific journals, and conference proceeding. The list of published papers is provided below.

1. Y. Yusof and M. H. Refai, "MMCAR: Modified Multi-class Classification based on Association Rule," in *Proceedings of the International Conference on Information Retrieval and Knowledge Management*, Kuala Lumpur, 2012, pp. 6-11.
2. Y. Yusof and M. H. Refai, "Modified Multi-Class Classification using Association Rule Mining," *Pertanika Journal of Science and Technology*, vol. 21, pp. 205-216, 2013
3. M. H. Refai and Y. Yusof, "Partial Rule Match for Filtering Rules in Association Classification," *Journal of Computer Science*, vol. 10, pp. 570-577, 2014.



CHAPTER ONE

INTRODUCTION

1.1 Background

In the field of Computer Science, data mining is one of the main phases in the Knowledge Discovery Database process (KDD). It involves the utilization of discovery algorithms and data analysis to produce particular details of patterns (or models) in the data under acceptable computational efficiency constraints [1-4]. The other phases in KDD are data cleansing, pattern evaluation, data reduction, data selection and visualization of the discovered information[2, 5]. In data mining, one of the main tasks studied is classification [5]. The main objective of classification is to create a model from a group of attributes where every attribute is the target class[6, 7]. This model is then used to forecast the classes of a new group of attributes [2, 8]. Classification has been applied in many areas, for instance medical analysis[9], space exploration[10] and textual mining [5, 8, 11].

Text classification (TC), has been one of the popular task in text mining [12-15], and it involves the understanding, recognition and organization of various types of textual data [16]. The objective of TC is to classify an incoming textual document into a group. The "supervised" learning classification classifies a new document on predetermined input text collection [2]. TC is a multi-phase process that includes processing of the textual documents, classifying the document based on an algorithm and evaluating the produced classification model [17]. A number of dissimilar classification methods are used in TC and these have been adopted from data mining and machine learning, for instance, decision trees [10, 18, 19], Naive bayes [20-22],

Support Vector Machine [23-25] and neural network [26]. These methods have mainly been investigated and used in the classification of English documents [15, 27, 28].

Alternatively, there is a method in TC that is known as Association classification (AC) [29, 30], which represents a field of research in data mining that combines association rules discovery with classification [10, 31]. The main objective of the AC is to build a model that is also known as classifier [32, 33], which consists of a specific amount of knowledge from labelled input, with the intending purpose of predicting the class attribute for a test data case that is accurate as possible [34]. AC is a promising data mining approach, which builds more accurate classifiers than traditional classification technique. This made by integrating association rules mining with classification.

The classifier is usually built based on the content of the training data set, and later been utilized to predict the category for new unseen document [19]. This type of learning is called supervised learning since the input data set contains labelled categories and the search for knowledge is restricted with target categories [24]. Multi class classification divides a training data according to class labels, for each class in the training dataset, rules are built in initially from the training items (depend the minimum support and minimum confident) [18]. One rule may be associated with multiple classes, but only the class with the largest occurrence will be considered by multi class AC methods.

In the last few years, several AC algorithms have been developed such as Classification Passed Association Rules CPAR [35], Live and Let Live (L3G) [36], Multi Class Association Rule MCAR [37], CACA [38], BCAR [39], LCA [31] and others. Previous studies have indicated that AC approach produces more accurate classifiers than others data mining approaches such as probabilistic [40], and decision tree [41]. AC unlike traditional data mining methods, such as neural network [42] and probabilistic methods [40], which produce classification models that are hard to understand or interpret by end-user, AC produces rules that are easy to understand and manipulate by end-users [37]. However, AC algorithms usually suffer from the exponential growth of rules which means they derive large number of rules which make the resulting classifiers oversized and consequently limit their use and it may be affiant to understand and maintain them.

Several AC techniques have been proposed in recent years such as CBA [43], CAEP [29], CMAR [30], Negative-Rules [44], CPAR [35], Negative-Rules [45], MMAC [46], 2-PS [116], MCAR [37], CACA [38], ACCF, and BCAR [39]. These techniques use several different approaches rules discover, rank rules, rule prune and rule prediction.

Data mining classification, there was only one data format in AC inherited from association rule mining called horizontal [43]. In the horizontal data format, created by the Apriori algorithm authors [47]. Apriori algorithm use multiple data scans when searching for frequent item sets which leads to high computation time. On the other hand, few association rule mining algorithms use the vertical format [48, 49]. The advantage of vertical data representation is when the cardinality of the

transactions identification (tid-list) becomes very large, intersection time gets larger as well. This happens particularly for large and correlated transactional databases.

Many rule pruning procedures have been employed in AC to reduce the size of the classifiers, some of which have been brought from decision trees like pessimistic estimation [50] and others from probabilities like Chi-square (χ^2)[51]. These pruning methods are used while building the classifier, for instance, an early pruning, which removes rule items that do not survive the support threshold, like database coverage [43]. The rule pruning is responsible about the classifier size (number of rule).

AC has deferent Class prediction or class assignment approaches [36, 37, 39], to appropriate class labels to test case, single rule prediction use one rule to apply in all the test cases, while the other methods use group rule prediction to test case, the single and group rule prediction use the higher rule ranked in the classifier to apply in the test case.

The AC approach uses association rule mining to discover the Classification Association Rules (CARs) [10]. The problems in AC during the discovery process, a large number of rules are produced. These rules were redundant especially when the support threshold that is used is very low [31]. Hence, the support threshold is the key factor, which controls the number of rules produced in AC. Based on that, the number of extracted rules are small if the support value is high. Thus, the rules are excluded with the high assurance rule, which may lead to discard the essential knowledge even though it will be helpful in the classification stage. Based on above, the support threshold is set to a very small value to solve this problem. Nevertheless, a large number of rules are generated, even though many of the rules are useless,

because they hold assurance values and low support. Several problems may occur due to the large number of rules such and this include over fitting [52, 53].

Furthermore, another problem in AC method is the removal of redundant rules is able to make the classification procedure more effective and accurate [54, 55]. It is not always helpful to get a large number of rules when classify a new test document since this may require long prediction time.

There is a great chance to have more than one rule contradicting each other in the answer class. In the data mining, the primary aim for classification is predicting the class labels of test cases, which can be classified into two main categories. The first category makes the prediction based on multiple rules [10, 35, 56], while the second category makes the prediction based on the highest precedence single rule applicable to the test case [43, 56, 57]. The problem of a single rule prediction is that it will specifically use when there is just single rule applicable to the test case, but there could be multiple rules applicable to the test case making the decision questionable. The main advantage of using multiple rules to predict are rules to contribute the class, which can limit the chance of every single rule for predicting all test the cases to satisfy the rules.

The main goal of this study is to develop a text classifier based on association classification. Particularly, the study presents an efficient method for discovering rules based on intersection sequence that requires only one database scan to generate rules from text collection. Furthermore, the need of progress a new rule pruning method to remove redundant rules (rules that lead to incorrect classification). And

later, develop a new prediction method that classifies unseen documents into correct categories.

1.2 Research Motivation

Association Classification (AC) is one the methods in data mining that has been effectively used to solve the real world categorization issues such as image processing [58], medical diagnoses [59, 60] and bioinformatics [61]. A number of experimental studies have shown that Association classification is a more accurate method in building classification model [43, 62, 63]. Furthermore, Association classification model produces rules, which may not be discovered using traditional classification algorithms SVM [23] KNN [22] C4.5 [41]; rule sample was attached in appendix C. The AC generates “If-Then” rules which are more comprehensible and controllable for end-users [63, 64].

The applicability of AC classification approach is mainly due to several advantages offered by this approach such as the simplicity of the produced classification model (classifier), a high prediction accuracy and the easy maintenance of the classifier where rules can be easily sorted, added and detached [8, 65].

1.3 Research Problem

AC approach usually produces more accurate classifiers than classic classification data mining approaches [36, 37, 39]. It is a data mining approach that have been studied extensively in the last decade and applied in various real world application domains including medical diagnoses [66], market basket analysis [67], security [68] and others. However, for unstructured textual data, AC mining has not yet

being thoroughly explored due to the representation complexity and the high dimensional of textual data. Therefore, mining unstructured and high dimensional data sets like Reuter [69] while using AC approach is a challenge.

One common shortcoming of most existing AC algorithms [31, 46, 70-73] is the massive number of rules produced by the classifier. In particular, the number of rules that might be generated in an association rule mining phase could reach thousands if not tens of thousands [27, 46, 74, 75]. Such a huge number becomes impractical and thus can limit their use in applications such as medical diagnoses and text categorization. This is due to utilized rule pruning methods such as database coverage [43] or lazy pruning [34]. Examples of AC that implement these methods includes the CPAR [35], CBA [43] and MCAR [37]. These AC classifiers produce either a high accuracy but with large size of classifier or a small set of rules that generates low classification accuracy. The MCAR [37], in particular, obtained such result due to three factors; a single form of data representation (i.e vertical layout), a rigid rule pruning method that relies on both the precedent and antecedent of a rule, and a single rule prediction.

This study discusses means on improving the MCAR by proposing a rule discovery method that reduces dimensionality through data representation. This is later complimented by a pruning method which eliminates redundant rules. We also provide a procedure of multiple rule prediction that enhances the classification accuracy while the number of rules is minimized.

1.4 Research Questions

Based on the issues highlighted above, this research will seek to answer the following questions:

- How can existing AC methods be improved in order to produce less number of rules without reducing the prediction accuracy?
- Will reducing the number of rules have an impact on the model's accuracy?
- Will a change in the class assignment methodology contribute to the prediction rate of the classification model?

1.5 Research Objectives

The main goal of this research is to develop a text classifier based on multi-class association rule mining. In order to achieve such goal, this research needs to accomplish the following:

- To design a rule discovery algorithm based on intersection of transactions identification (TID-list), represented using vertical and horizontal data layout to reduce data scanning, which in turn reduces computational time.
- To design a rule pruning algorithm based on database coverage pruning method that eliminates redundant rules and reduces number of rules.
- To design a group based class prediction algorithm that enhances classifier accuracy.

1.6 Research Scope

The scope of this research focus on both structured and unstructured textual collections. For unstructured data set, the Reuters-21578, which is a commonly used text collection in data mining [69] that contains 21578 Reuter's news documents is utilized. The news documents in this data set are about different subjects such as 'people', 'places' and 'topics'. The total number of categories is 672. For the structured data sets, we have selected fourteen data sets from UCI data repository [76] that have different number of training cases and attributes.

The proposed AC algorithm is evaluated by comparing its results with existing AC and rule-based classification algorithms such as C4.5, MCAR and PART. The bases of the comparison are textual evaluation measures test that includes accuracy and the number of rules for the structured data sets (UCI data), as well as the unstructured data set (Reuters).

1.7 Research Contributions

A summary of the contributions of this research are as follows:

1. An AC data mining algorithm that operates on structured and unstructured data.
2. A rule discovery algorithm that uses TID-list intersection to reduce time for create frequent rule item.
3. A rule pruning algorithm that reduces the size of the classifier.
4. A rule prediction algorithm that improves classification accuracy by developing new group of rules if relevant.

1.8 Thesis Organization

This thesis is organized into 6 chapters including this chapter. The following paragraphs provide brief descriptions of the remaining chapters of this thesis.

Chapter Two includes literature review on the classification and association rule discovery approaches in data mining. It also discusses various approaches used by AC to discover frequent item, rule pruning and prediction method.

Chapter Three presents the research methodology of this research, which describes the methods and tools that were used in this research, moreover, describes the experimental phases for the performance association classification algorithms.

Chapter Four presents the proposed Modified Multi Class Association Rule (mMCAR). The chapter should illustrate detailed experimental on UCI data collection and Reuters. Through this chapter will determine all the steps of experimental, and identify clearly the methods that will be used in the experimental.

Chapter Five Presents the results for the experiments conducted for structured and unstructured data.

Structured data use UCI data sets, for unstructured Reuters-21578 dataset. The evaluation measure will be number of accuracy number or rule and computational time and breakeven point.

Chapter Six, Provide a summary of the entire thesis, including the research contribution and gives suggestions for the direction of possible future works with regards to this research.

CHAPTER TWO

RELATED WORK

2.1 Introduction

This chapter introduces the essential concepts related to the concerns of this research as well as discusses related work relevant to this research. In doing so this chapter will focus on the following six main categories; In first section, data pre-processing will identify the methods to process the unstructured data. In text classification section, will introduce the most popular approaches, then will be describe the techniques that used to assess the classifier in the section of evaluation measures in text classification. Furthermore, Association classification section reviews association classification approaches which include association rules btechniques, pruning methods in Association classification and the prediction methods.

2.2 Data pre-processing

The first process in TC involves the transformation of documents which are called unstructured data; this is done by doing a bag word [77, 78] which removes any unnecessary data and numerical vector representation [79] to represent the data numerically, to makes it suitable for the learning algorithm and classification task. The next two sub-sections discuss these two methods.

2.2.1 A Bag of Word Representation

The most basic way of representing documents in a structured form is through representing documents in a structured form is through the “bag of words” method.

A ‘bag of words’ is a set of words which are not ordered and inconsistent in their size, depending on the length of a given document. A bag of words is denoted by $D_{jc} = \{W_{j1}, W_{j2}, W_{j3}, \dots, W_{jn}\}$ where j is an index of a particular document and n is the size of the set of words. The transformation of a document into a bag of words involves three steps [77]. The first step is called Tokenisation, whereby the document is segmented into tokens by white space. The second step, called stemming involves the conversion of each token into its root (stemming). For example, plural nouns are changed into singular forms and past tense verbs are changed into their root. Finally, the last step called stop word elimination, involves removing words which perform just grammatical functions regardless of the content of the text. These words contain prepositions, conjunctions, particles, auxiliary verbs, and so on. When these three steps are completed, a structured list of words representing a document is produced.

2.2.2 A Numerical Vector Representation

When an unstructured document is represented in structured form, each distinct word corresponds to a feature and the number of times a particular word occurs in the document corresponds to its value. The words, which are selected as features of the numerical vectors, are denoted by W_1, W_2, \dots, W_n . However, this representation scheme results in very high-dimensional feature spaces which may contain 10,000 dimensions or more[79]. Thus, in order to avoid unnecessarily large feature vectors, researchers have suggested several methods for feature selection, such as mutual information[80], chi-square[80], frequency based method [80], and information theory based method[81].

Three common methods are used to define feature values in numerical vectors. The first method uses a binary value that is, zero or one, for each word. Zero indicates the absence of the word while one indicates the presence of the word in a given document. The second method uses the frequency of every word in each document as its value. In this way, each element is an integer. The third method uses the weight of each word, w_k ($1 \leq k \leq n$), as a feature value, based on any term weighting method such as Inverse Document Frequency (IDF) or Weighted Inverse Document Frequency (WIDF) [82, 83]. The next two sub-sections will discuss the nature of term weighting approaches and feature selection methods.

2.2.2.1 Term Weighting

Term weighting, an important issue in TC, has been widely investigated in information retrieval (IR) [51, 84, 85]. Term weighting refers to value given to a term in order to reflect the significance of that term in the document.

A. Term Frequency

Tokunaga [82] defined Term Frequency as the simplest term weighting methods which is applied to examine the significance of each term in a given document. It is assumed that by using TF method, the term has a value relative to the number of times occur. Usually, for a term t and a document d , the following equation is used to calculate the weight of t in d :

$$W(d, t) = TF(d, t) \quad (2.1)$$

According to Rijsbergen [86] that the TF method can be used to enhance the TC and IR assessment measure called recall. Recall is represented in the form of an equation (2.13).

B. Inverse Document Frequency

While TF reflects the importance of a term in a single document, however, it does not reflect the occurrence of a term in a set of documents. In this regard the Inverse Document Frequency method (IDF) offers a solution. Inverse Document Frequency (IDF), means the importance of each term is inversely proportional to the number of documents that contain that term.

Equation 2.2 defines the IDF when the term t is within n documents as following:

$$IDF(t) = \log(N/n) \quad (2.2)$$

Slaton combined the TF and IDF [87] in terms of mass and this approach resulted in better performance when compared to other techniques. The combination of product TF IDF is given in equation (2.3)

$$W(d, t) = TF(t).IDF(t) \quad (2.3)$$

C. Weighted Inverse Document Frequency

One of the weaknesses of IDF is Binary Counting [20]. Binary counting treats each term in the documents equally. This weakness can be overcome through the Weighted Inverse Document Frequency (WIDF) algorithm. The WIDF of a term t in a document d is given as:

$$WIDF(d, t) = \frac{TF(d, t)}{\sum_{i \in D} TF(i, t)} \quad (2.4)$$

Where $TF(d, t)$ is the occurrence of t in d , and i indicates the range over the documents in the collection D . However, during the collection to the normalized term frequency, WIDF weight of a term is given as:

$$W(d, t) = \text{WIDF}(d, t) \quad (2.5)$$

Based on that, it is pertinent to note that studies on the above mentioned term weighting approaches have indicated that these approaches have produced good results when used on English text collections [20, 82, 88].

2.2.2.2 Feature Selection and Dimensionality Reduction

This procedure refers to a process of selecting the best K terms as a subset of the terms occurring in the training set and using only this subset as features in TC. This procedure results in the achievement of two main goals. Firstly, it trains more efficiently through decreasing the high dimensionality of the effective vocabulary. Secondly, it frequently increases classification correctness by removing rare terms[20]. There are many attribute selection methods available such as Document Frequency (DF), Information Gain (IG) [89], and Chi-square Testing (χ^2) [90]. The following sub-sections will discuss these methods.

A. Chi-square Testing

Snedecor [90] stated that the chi-square testing (χ^2) is “a well-known discrete data hypothesis testing method in statistics”. It determines if the variables are correlated or independent by assessing the correlation between two variables. The independence test, implemented to a population of subjects will determine whether the variables are negatively correlated or vice versa. Thus, the following equation (2.6) can be defined by the χ^2 value for each term t in a category c accordingly:

$$X^2(t, c) = \frac{N(AD - CB)^2}{(A + C)(B + D)(A + B)(C + D)} \quad (2.6)$$

Where C is the number of documents in c not containing t , the total number of training documents is N . A refers to the number of documents in c containing t , and B is the number of documents not in c containing t , while D is the number of documents not in c not containing t . χ^2 . This test was used by Yang and Padersen [80] in TC and showed promising results.

B. Information Gain

Information Gain (IG) is a method commonly used to measure goodness in machine learning. The goodness value refers to the amount of information gained when a prediction is conducted with the presence or absence of a term in a document. measure in the field of machine learning [89, 91]. It measures the amount of information gained for category prediction as a function of the presence or absence of a term in a document. IG is formulated in the following equation: 2.7.

$$IG(t) = - \sum_{i=1}^m P(c_i) \log P(c_i) + P(t) \sum_{i=1}^m P(t, c_i) \log P(t, c_i) + P(\bar{t}) \sum_{i=1}^m P(\bar{t}, c_i) \log P(\bar{t}, c_i) \quad (2.7)$$

- In this equation m is the number of categories, $P(c_i)$ is the probability of the category c_i .
- $P(t, c_i)$ is the joint probability of the category c_i and the occurrence of the term t .
- $P(t)$ is the probability that the term t occurs in a document, and $P(\bar{t})$ is the probability that the term t does not occur in a document.

2.2.2.3 Document Frequency

The Document Frequency (DF) method basically measures how many documents contain a particular word. To do this, it is necessary to compute the DF for each unique term in the training documents and remove those terms whose DF is lower than a predetermined threshold. The selection of frequently occurring words will improve the chances that these features will be presented in future test cases. In a study carried out by Yang and Pedersen[80], it was shown that DF performance is better than Mutual Information. However, it is often dominated by IG and Chi square measures.

The previous sections have discussed the advantages and disadvantages of the various procedures available in data mining. The next section will proceed to review learning Approaches to text categorization.

2.3 Text Classification

In the field of data mining and machine learning a number of Text Classification (TC) approaches exist., These approaches include Decision Trees [41], Support Vector Machine [92], Naïve Bayes [40], and Neural Network [26]. In the next sub-sections, TC methods will be discussed.

2.3.1 K-nearest Neighbor

KNN is a statistical classification approach, which has been intensively studied in prototype detection for over four decades. KNN has been successfully applied to TC problems [93-95] and shows promising results when compared to other statistical approaches such as Bayesian based Network [6, 96].

The KNN algorithm is quite simple. In a given situation of training and testing documents, the KNN algorithm will proceed to find the k -nearest neighbours amid the training documents, as well as uses the categories of the k -neighbours to weight the category of the test document. The evaluation scores of every neighbour document to the test document are used as a weight of the categories of the neighbouring document. If many k -nearest-neighbours share a category, then the pre-neighbour weights of that category are added together, and the resulting weighted sum is used as the probability score of that category with respect to the test document. By sorting the scores of the candidates' categories, a ranked list is obtained for the test document [80, 93].

Actually, the value of K is fixed beforehand in the traditional k NN algorithm, while, the big classes will overwhelm small ones if k is too large. According to Jirng [97] refers that the advantage of k NN algorithm, which could make use of many experts, will not be exhibited k is too small, actually, could make use of many experts. Other issues in k NN are similarity and distance measures, computational complexity, dimension reduction feature selection [98]. K NN requires more time for classifying objects when a large number of training examples are given. K NN should select some of them by computing the distance of each test objects with all of the training examples. Practically Othman [99] compared five algorithms of classification using breast cancer dataset. The mean of total Error for K -nearest Neighbour was 32.3840% and Root Relative Squared Error 79.496%, the time taken to create the classifier 0.81 second.

2.3.2 Naive Bayesian

As indicated by Thabtah [20] and Hadi [100] Naive Bayesian is a simple probabilistic classifier based on applying Baye's theorem [101]. It is a predictive, easy and language independent method [102, 103].

In the study conducted by Othman [99] five algorithms of classification using breast cancer dataset were compared., Using the Bayesian classifier, the Mean Absolute Error for NB was 22.2878 % and Root Relative Squared Error 65.1135 %. The time taken to create the classifier is 0.19 seconds.

NB classifier is not very robust to classify noise since independence of the attributes is not preserved. problem related to NB is its inability to classify noise [104].

2.3.3 Decision Trees

According to Quinlan [41], the most popular decision tree learning program is C4.5. This approach begins by selecting the best attribute as a root node, where each branch of the root corresponds to one of its possible value. The process is then repeated on each branch until no examples are left in the training data set. In order to decide which attribute is to be selected at each step, information gain (IG) is used [89]. The attribute with the highest gain is chosen as the node. In an informal situation, IG measures how good an attribute separates the training set with respect to the class labels. Therefore, the higher the gain, the better the separation resulting from classifying training examples on the associated attribute. In a formal situation, IG provides equations for computing information gain Joachims [79] Mitchell[91] applied C4.5 and other TC methods in two data sets, and the results showed that the

C4.5 procedure produced competitive results if compared with other methods such as KNN[93], SVM[92] Rocchio [105].

In the study conducted by Othman [99] the Mean Absolute Error for Decision Tree was 39.2681% and Root Relative Squared Error 73.57%. The time taken to create the classifier 0.23 seconds.

In classification, while the aim of reducing the error rate to zero, requires a long training phase, which may deteriorate in general performance of the resulted classifier on test data objects. The general description of the over fitting problem, which can occur due to many reasons such as a noise among the training objects or limited number of training data objects [106]. In decision tree algorithms for instance, it is possible to construct a highly accurate decision tree for the training data, but, during the construction of the tree it is usually useful to stop the building process early in “order to generalise the performance of the outcome on test data objects. Therefore, pruning approaches like pre-pruning and post-pruning [18] have been widely used during building decision trees” in order to provide accurate performance on test data and to avoid over fitting the training data very well.

2.3.4 Support Vector Machine

Support Vector Machine SVM was introduced by Vapnik, [92] as a class of supervised machine learning techniques. It is based on the principle of structural risk minimisation. In linear classification, SVM creates a hyper plane that separates the data into two sets with the maximum-margin. A hyper plane with the maximum-margin has the distance from the hyper plane to points when the two sides are equal.

In mathematical terms, SVMs learn the sign function $f(x) = \text{sign}(wx + b)$, where w is a weighted vector in \mathbb{R}^n . SVMs find the hyper plane $y = wx + b$ by separating the space \mathbb{R}^n into two half-spaces with the maximum-margin. Linear SVMs can be widespread for non-linear problems. To do so, the data is mapped into another space H and the linear SVM algorithm is performed over this new space. In recent times SVM has been successfully used on TC [79] and they produce better results with reference to accuracy when compared to other machine learning techniques such as NB, decision trees, and KNN. support vector machine (SVM) problems the high-dimensional classification [107].

There are main two issues in SVM. First, it is applicable to only binary classification. If a multiple classification problem is given, it must be decomposed into multi binary classification problems using SVM. The second issue is the sparse distribution, representing documents into numerical vectors, training examples generates zero values very frequently and since inner products of its input vector [108].

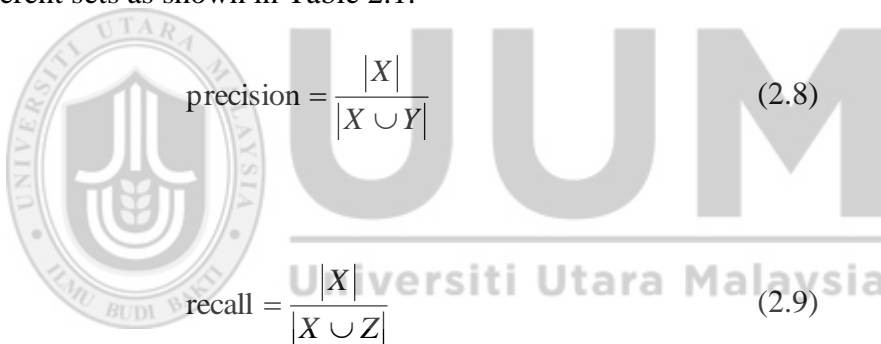
2.4 Evaluation Measures in Text Classification

Most existing TC techniques use the popular error-rate method [42,149,150] to estimate the effectiveness of their classifiers. Basically, the classifier predicts the class of a test data case, which is an error if not counted, and right or success if it is counted. However, the overall error on the data is gotten when the total number of cases in a test data is divided into number of error cases. Hence, prediction accuracy is measured by the error-rate of a classifier on a test data set.

Precision is a method for evaluation used in TC. Moreover, in the IR field, the precision was presented together with the Recall method [86]. Recall and Precision work as follows: as the first step, they have a query through a collection of objects/documents. Several of the objects relate to the query and some of other objects are not. It can make two kinds of mistakes that are false negatives and false positives. Precision measures the proportion of correct answers from all those that were retrieved while recall measures the proportion of correct answers retrieved from the set of all correct answers.

Generally and with respect to a given query, documents can be divided into four different sets as shown in Table 2.1.

and



$$\text{precision} = \frac{|X|}{|X \cup Y|} \quad (2.8)$$

$$\text{recall} = \frac{|X|}{|X \cup Z|} \quad (2.9)$$

Table 2.1
Documents possible sets based on a query in IR

Iteration	Relevant	Irrelevant
Documents Retrieved	X	Y
Documents not Retrieved	Z	W

In the case of classification problems in data mining, precision functions similarly to accuracy and problems can be described in terms of class by class or globally. However, to derive precision, the number of cases classified in the class can divide the number of correct classifications, within each class. In the test, the total number

of cases set can divide the number of correct classifications, which will refer to the precision. Nevertheless, recall method shows better results when is performed class-by-class. For example, the number of cases divides the correct classifications, which should have been classified in that class to obtain recall.

Provost and Kohavi [109], proposed a universal method called confusion matrix in the case of binary classification that count the cost of wrong prediction. Confusion matrix is like a precision and recall methods in that it consist of information about actual and predicted classifications carried out by the classifier. The performance of the resultant classifier is commonly evaluated using the data in the matrix.

Table 2.1 illustrates a confusion matrix where “Documents Retrieved” and “Relevant” represent the class “yes” while “Documents not Retrieved” and “Irrelevant” represent class “no”. In Table 2.1, “X” corresponds to what is so called true-positive and represents the number of cases when the predicted outcome matches the actual class for class “yes”. “Y” represents the outcome when it is incorrectly predicted as “no” when it is in fact “yes” and is called false-positive. “Z” represents the outcome when it is incorrectly predicted as “yes”, when it is in fact “no” and is called false-negative. Lastly, “W” is known as true-negative and represents the number of cases where the predicted outcome matches the actual class for “no”. The accuracy on a data set can be obtained by adding the values of “X” and “W” from the confusion matrix of that data set.

On the whole, TC researches, including those carried out by [20, 39, 79, 93] use the error-rate (accuracy) method. In addition through using Precision, Recall, and F1 the effectiveness of their classifiers is improved.

2.5 Association Classification

Association Classification (AC) is based on the association of rules. It is an integration of the two significant data mining tasks, namely, association and classification. The AC approach works in the following way. Firstly, all rules that satisfy user-specified restrictions (*minsupp*, *minconf*), are produced using an association rule mining algorithm. However, since the number of rules generated run into several thousands, and furthermore many of them are both redundant and not discriminative among the classes, they need to be pruned using pruning procedure(s). At this stage a number of rules can be reduced. The rules that are left are the interesting ones that will form a model (classifier) used to classify new data. However, each one of the classifiers should have a defaulting rule which is useful when no classifiers rule preserve to be used.

If ruleitems satisfy *minsupp* they are said to be frequent ruleitems. In general, the item has passes *minsupp* in association rule mining will be known as a frequent item set. If the frequent itemset consists of only a single attribute value, it is said to be a frequent 1-itemset.

Table 2.2
Training Data

Row#	AT1	AT2	Class
1	Z1	W1	P1
2	Z1	W2	P2
3	Z1	W1	P2
4	Z1	W2	P1
5	Z2	W1	P2
6	Z2	W1	P1
7	Z2	W3	P2
8	Z1	W3	P1
9	Z2	W4	P1
10	Z3	W1	P1

For example, with a *minsupp* of 20%, the frequent 1-itemset in Table 2.1 are $\langle (AT1, z1) \rangle$, $\langle (AT1, z2) \rangle$, $\langle (AT2, w1) \rangle$, $\langle (AT2, w2) \rangle$ and $\langle (AT2, w3) \rangle$. A ruleitem is a combination of itemsets and a class label in the form $T_1, T_2 \dots T_m \rightarrow c$ where T_i is a set of itemsets and c is a class. If a ruleitem is passing confident, then it is called an accurate ruleitems.

It is the case that at present, AC methods generate frequent ruleitems by scanning a few times over the training data set. However, with the first time scan will identify the support of 1- ruleitems then to be frequent in the previous scan they start with the ruleitems that are found, so as to generate new possible frequent ruleitems that increase more attribute values. In other words, frequent 1- ruleitems is used to discover frequent 2- ruleitems, and frequent 2- ruleitems is the input for the discovery of frequent 3- ruleitems and so forth. After all the frequent ruleitems have been discovered, based on algorithms of association rules, the process of

classification is carried out. Thus, extracting a complete set of class-association-rules (CAR) from those frequent ruleitems that excused the minconf threshold.

Liu [43] proposed one of the first algorithms to combine association rules with classification. In fact, there are two main phases in the process, Apriori algorithm [47] is implemented in phase one to discover frequent ruleitems, and stage two involves building the classifier. Experimental results indicate that the approach developed in [43] produced rules which are competitive to popular learning methods like decision trees [41].

It has been pointed out that when classification and association rule mining are combined, it is possible to produce efficient and accurate classification systems[43, 46] . This is evidenced by the fact that a number of empirical studies have shown that AC is often capable of building more accurate classification systems compared to traditional classification techniques. [34, 43, 46, 52, 110, 111]. In addition, AC create rules being easy to recognize and influence by end-users, unlike neural network and probabilistic approaches which are produce classification models that are hard to understand or interpret by end-users, when the association classification models generates IF Then rules which are more comprehensible and controllable for end user. However, one shortcoming of AC algorithms is that they have been investigated mainly on classic classification benchmarks such as UCI Archives [76], which are simple and medium sized data sets. In other words, AC has yet to be applied on large and complex data collections such as TC in order to evaluate its effectiveness and efficiency. Thus, one of the ultimate objectives of this thesis is to extend existing AC approach on large and unstructured data collections (TC).

Several AC techniques have been proposed in recent years, such as CBA [43], CAEP [29], CMAR [30], ARC-AC [44], CPAR [35], MMAC [46], 2-PS (Qian et al.,2005), MCAR [37], CACA [38], ACCF [112], BCAR [39], and ACN [45]. These techniques use several different approaches to discover rules, prune redundant and classify new test cases. The next sub-sections present a survey of common Association algorithms in data mining.

2.5.1 Classification Based on Association Rule

One of the earliest studies that illustrated the utilisation of association rule in classification benchmarks is Classification Based on Association Rule (CBA)[43]. The CBA implements the Apriori algorithm [47] to discover frequent ruleitems. This stage is called candidate generation. The frequent ruleitems are (<attributes, values>, class) that exceeds *minsupp*. Then, these frequent ruleitems are used to produce the complete set of CARs, which are then used to form the classifier. This stage is called classifier building.

There are various processes involved in candidate generation (Apriori algorithm) and the classifier building steps of CBA. In the candidate generation stage, the search for frequent 1- ruleitems is first carried out following which the disjoint frequent 1- ruleitems are combined to form candidate 2- ruleitems. This process is repeated until no more frequent ruleitems can be found. In fact, CBA focuses on a special subset of association rules whose right-hand-side is restricted to the class attribute. The CARs are rules whose consequents are limited to the class label in a form $A \rightarrow c_i$ where A is an attribute value and c_i is a possible class.

Furthermore, the CBA classifier is created when CARs are first produced following which, a subset is chosen to form the classifier. The algorithm first ranks all the derived CARs according to the ranking procedure. The rule gets an opportunity to be inserted into the classifier if it correctly covers at least one training data case. If a rule is inserted into the classifier, all cases inside the training data that are covered by the inserted rule are removed. This process is stopped when all training data cases are covered by some rules or all candidate rules are used. When this happens, the majority class among all cases left in the training data is selected as the default class.

The CBA uses only a single minimum support in rule generation, which is inadequate for unbalanced class distribution, and classification data often contains a huge number of rules, which may cause combinatorial explosion. For many datasets, the rule generator is unable to generate rules with many conditions, while such rules may be important for accurate classification.

2.5.2 Multi-class Classification based on Association rule

The Multi-class Classification based on Association rule (MCAR) [37] focuses on the rule ranking scheme which ensures that rules with high assurance are kept for prediction. MCAR consists of two major stages, namely, rules production and classifier building.

In the first stage, the training data set is scanned once in order to discover frequent *one-ruleitems*, and then MCAR combines the *ruleitems* generated to produce candidate *ruleitems* that involve more attributes. Any *ruleitem* with support and

confidence larger than *minsupp* and *minconf*, respectively, is created as a candidate rule. In the second stage, the rules created are used to build a classifier based on their effectiveness on the training data set. Only rules that cover a certain number of training cases are kept in the classifier.

MCAR continues with two type of data, integers and real [113]. Through the scan, frequent *1-ruleitems* are determined, and their occurrences in the training data (rowIds) are indexed inside an array in a vertical format. In addition, classes and their frequencies are indexed in an array. Any *ruleitem* that fails to pass the support threshold is discarded. MCAR drive the produced function to locate frequent *ruleitems* of size k by appending disjoint frequent itemsets of size $k-1$ and intersecting their rowIds. Based on above, the frequent *ruleitems* detection method in work by MCAR scans the training data set for counting the frequencies of *1-ruleitems* in order to determine those that hold enough support.

Furthermore, the result of a simple intersection between rowIds of two itemsets gives a set, which holds the rowIds where both itemsets happened together in the training data. This set along with the class array, can hold the class labels of frequencies and is used during the first scan, can be created to count the support and self-confidence of the new *ruleitem* that results from the intersection. The produce function is increase iteratively for every set of frequent itemsets produced at iteration K in order to produce probable frequent *ruleitems* at iteration $K+1$. As [114] [36] point out, since the number of rules generated by AC can be large, it is important to select a suitable rule set for forming the classifier.

Generally, in AC, the rule ranking is based on the cardinality of the rule's antecedent, support, and confidence. The advantage of MCAR is that it contributes further to previous rule ranking approaches by looking at the class distribution frequencies in the training data and prefers rules that are associated with dominant classes.

The strength of MCAR is its ability to generate rules with multiple classes from data sets where each data objects is associated with just a single class.

2.5.3 Classification based on Multiple Association Rules

According to Li [30], another AC algorithm that selects and analyses the correlation between high confidence rules, instead of relying on a single rule is the Classification based on Multiple Association Rules CMAR algorithm. It stores rules in a prefix tree data structure known as a CR-tree. The CR-tree store all rules in a descending order depend of the rule frequency of their attribute values appearing in the rule occurred. The first rule is generated; it will be inserted into the CR-tree as a path from the root node. Its support, confidence and class are stored at the last node in the path. When the second rule is inserted into the tree and it contains common features with another existing rule in the tree, the path of the existing rule is extended to reflect the addition of the new rule.

The CMAR uses a set of related rules to make a prediction decision by evaluating the correlation among them. The CMAR algorithm adopts the chi-square testing in its rules discovery step. When a rule is found, CMAR tests whether its body is

positively correlated with the class. If a positive correlation is found, CMAR keeps the rule, otherwise the rule is discarded.

In addition, a new prefix tree data structure called CR-tree which handles the set of rules generated and speeds up the retrieval process of a rule has been introduced. The CR-tree has proven to be effective in saving storage since many condition parts of the rules are shared in the tree. Experimental tests using CMAR [30], CBA [43] and C4.5[18] on different data sets[76] have shown that the classifiers generated by CMAR are more accurate than those of C4.5 and CBA on 50% of the benchmark problems considered. Furthermore, the results revealed that 50%-60% of space can be saved in the main memory using the CR-tree when compared to CBA.

Scan the training data two times will give the time consumed to find the complete set of rules that meet certain support and confidence thresholds, and then it scans the training data set again to construct an FP-tree.

2.5.4 Classification based on Predictive Association Rule

Another AC algorithm called Classification based on Predictive Association Rule Classification based on Predictive Association Rule (CPAR) is also available for data mining according to Yin and Han [35]. Quinlan and Cameron-Jones [18] stated that, CPAR adopts FOIL in generating the rules from data sets, to find the best rule condition that generates the biggest gain between the available ones in the data set. It condition is identified; the weights of the positive examples associated with it will be deteriorated by a multiplying factor. This procedure will be repetitive until all positive examples in the training dataset are covered. The searching process for the

best rule condition is the largely time consuming process of CPAR since the gain for each possible item wants to be calculated in order to find out the best item gain. In the rules generation process, CPAR derives not only the best condition but also all similar ones since there are often more than one attribute item with similar gain. It has been claimed that CPAR improves the efficiency of the rule generation process if compared with popular methods such as CMAR [30] and CBA[43].

The CPAR hence generates and tests more rules than traditional rule-based classifiers to avoid missing important rules, and uses expected accuracy to evaluate each rule and uses the best k rules in prediction to avoid overfitting.

2.5.5 Multi-class, Multi-label Association Classification Approach

According to Thabtah [46], the MMAC algorithm is considered the only multi-label algorithm in AC. It consists of three stages, namely, rules generation, recursive learning and classification. In the first stage, it scans the training data to discover a all set of CAR. Training cases that are associated with the CARs that are produced are discarded. At second stage, MMAC will carry on to discover more rules that pass the *minsupp* and *minconf* thresholds from the remaining unclassified cases, until no further frequent ruleitems can be found. At last, rule sets derived throughout every iteration are merged to form a global multi-label classifier which is referring to tested against test data.

The results obtained from 28 different data sets have indicated that the MMAC approach is precise and is an efficient classification method, and is highly aggressive and scalable in evaluation with the other customary and AC approaches

like PART, RIPPER, and CBA [43]. The MMAC is its ability to generate rules with multiple classes from data sets where each data objects is associated with just a single class.

2.5.6 Two-Phase Based Classifier Building

Qian [115] approach first builds a classifier through a 2-PS (Two-Phase) method. The first phase aims to prune rules locally, that is, rules mined within every category are pruned by a sentence-level constraint. This makes the rules more semantically correlated and less redundant. In the second phase, all the remaining rules are compared and selected from a global perspective, which means training examples from different categories are merged together in order to evaluate these rules. In addition, when predicting a new document, the multiple sentence-level appearances of a rule are taken into account. Experimental results on the well-known text corpora Reuters-21578 [69], have shown that the 2-PS algorithm achieved a higher accuracy than many well-known methods such as SVM, KNN, C4.5 and NB. 2-PS algorithm there is extensive on how to determine the optimal number of components.

2.5.7 Class Based Association Classification

In Tang and Liao [38], a new class-based AC approach called Class Based Association Classification CACA was proposed. CACA first scans the training data set and stores data in the form of a vertical format like MCAR [37]. After that it calculate the frequency of each attribute value and arranges attributes in descending order depend to their frequency. Any attribute have fails to pass the *minsup* is removed at this stage. The staying attribute values are then tested for intersect attributes depend on class in order to cut down the searching space of frequent

patterns. Every attribute in a class group that passes the *minconf* threshold, is inserted in the Ordered Rule Tree (OR-Tree) as a path from the root node, and its support, confidence and class are stored at the last node in the path. CACA classifies the unseen data in the same way like CBA. Experimental results suggest that CACA performs better with reference to accuracy and computation time than MMAC on UCI data sets. CACA uses only a single minsup in rule generation, which is inadequate for unbalanced class distribution, number of rule so big.

2.5.8 Association Classifier with Negative Rules

An AC with negative rules (ACN) was proposed by Kundu [45]. ACN extends the Apriori algorithm to mine a relatively large set of negative association rules and then uses both positive and negative rules to build a classifier. A positive rule takes the form of $X \Rightarrow Y$ where X, Y are a set of items and $X \cap Y = \emptyset$ while a negative rule takes the form of $X \Rightarrow Y$ where in addition to being a set of items, X or Y will contain at least one negated item. ACN builds a classifier similar to CBA but generates the rules in a different way compared to CBA. The negative rules will be generated in all phases of the Apriori candidate generation procedure based on the least rule items.

These negative rules will not take part in the generation of any new rule but they will compete for a place in the final classifier with the positive rules. Results from the experiments [45] show that ACN is not only time efficient but also significantly better than three other classification methods, that is CBA, CMAR, and C4.5 with respect to accuracy when applied to the UCI data sets [76].

2.5.9 Association Classification based on Closed Frequent Itemsets

Another procedure proposed by Li [112], is the ACCF. In this procedure, an *Itemset* "X" is a closed frequent *Itemset* in a data set S if there is no proper super-*itemset* Y, such that Y has the same support count as X in S, and X satisfies *minsupp*. This method is an extension of an efficient closed frequent pattern mining method called Charm to discover all frequent closed *itemsets* (CFIs) [116]. This would help in the generation of the CARs. The results obtained from experiments on 18 data sets from UCI repository showed that ACCF is consistent and highly effective at classifying various kinds of data sets and has a better average classification accuracy in comparison with CBA [43].

2.5.10 Boosting Association Rules

A method called BCAR was developed by Yoon and Lee [39], in which a huge amount of association rules are produced. Then, the rules derived are pruned "using a method equivalent to a deterministic Boosting algorithm[117]. This pruning method is a modification of the database coverage pruning [43]. The BCAR algorithm can be utilized in a large-scale classification benchmarks such as TC data. Experiment using a variety of text collection show the BCAR achieves good prediction if compare with SVM [92] and Harmony[30].

2.5.11 Association Classification based on Compactness of Rules

In Niu [118], ACCR was proposed, which extends the Apriori algorithm to generate classification rules. This would help overcome the twin problems when on the one hand, many good quality rules will be ignored when the user sets the support threshold too high, and on the other hand, too many redundant rules will be

generated when the support value is set too low, which consequently consumes more processing time and storage [118]. Consequently, Niu [118], developed a metric measure of rules called "compactness" that stores rule items with low support but high confidence, which ensures that high quality rules are not deleted. The compactness is computed as follows:

$$Compactness(I) = \frac{\sum_{i=1}^m Lift(R_i)}{m} \quad (2.17)$$

$$R_i = (I - \{I_i\}) \rightarrow I_i \quad (2.18)$$

$$Lift(X \rightarrow Y) = \frac{Conf(X \rightarrow Y)}{Sup(Y)} = \frac{Sup(X \cup Y)}{Sup(X)Sup(Y)} \quad (2.19)$$

Where, the "lift" is the degree of independence between antecedent items (X) and consequent items (Y) of the measured rule $X \rightarrow Y$. If the value is close to 1, the relationship between antecedent and consequent is small. ACCR builds a classifier similar to CBA. The experimental results obtained from tests on UCI data sets illustrated that the ACCR algorithm has better accuracy in comparison with CBA and CMAR algorithms.

2.6 Rule Discover and Production in Association Classification

This section explains the association rules techniques used to discover items, which include three methods, such as, Apriori, Frequent pattern growth and Tid-list intersection.

2.6.1 Apriori

Agrawal and Srikant [47] proposed an algorithm called Apriori, which is based on the fact uses the prior knowledge of frequent itemsets. As mentioned earlier in (CBA), the discovery of frequent itemsets is accomplished in a step by step fashion, where in each iteration, a full scan over the training data is required to generate new candidate itemsets from frequent itemsets already found in the previous step. Apriori uses the “downward-closure” property, aiming to improve the efficiency of the search process by reducing the size of the list of candidate itemsets during each iteration.

CBA [43], CAN [45] are algorithms that implements the Apriori algorithm to discover the frequent ruleitems and this step is called candidate generation. These frequent ruleitems are (\langle attributes, values \rangle , class) that pass minsupp. Then, the frequent ruleitems are used to produce the complete set of CARs, which in turn is used to form the classifier. This step is called classifier building.

```

“F 1 = {large 1-ruleitems};
CAR 1 = genRules (F 1 );
prCAR 1 = pruneRules (CAR 1 );
for (k = 2; F k-1 ≠ ∅; k++) {
C k = candidateGen (F k-1 );
for each data case d ∈ D
C d = ruleSubset (C k , d);
for each candidate c ∈ C d {
c.condsupCount++;
if d.class = c.class then
c.rulesupCount++;
}
F k = {c ∈ C k | c.rulesupCount ≥ minsup};
CAR k = genRules(F k );
prCAR k = pruneRules(CAR k );
}
CARs = ∪ k CAR k ;
prCARs = ∪ k prCAR k”

```

Figure 2.1: Candidate Generation Apriori Algorithm

Figures 2.1 depict the candidate generation (Apriori algorithm) and the classifier building steps of CBA. In the candidate generation phase, the search for frequent 1- ruleitems is first implemented, and then the disjoint “frequent 1- ruleitems are combined to form candidate 2- ruleitems. The process is repeated until no more frequent ruleitems can be found”. In fact, CBA utilises the association rule (Apriori) in discovering frequent ruleitems and focuses on a special association rules subset.

2.6.2 Frequent Pattern Growth

Apriori-like techniques use a candidate generation step to locate frequent itemsets during each iteration. Thus these techniques require significant processing time and memory. In this regard, Han [119], introduced a new association rule mining approach called FP-growth that generates a highly condensed frequent pattern tree (FP-tree) representation of the transactional database. Every database is appeared in the tree by just one path and the length of every path is same to the number of the frequent items in the transaction representing that path. The FP-tree is a helpful data representation because, every frequent itemsets in all transaction of the original database are given by the FP-tree, when there are a many of mach between frequent items, And the FP-tree need only two database scans, in the first, scan all frequent itemsets along by their support in all transaction are produced and in the second scan constructed the FP-tree method.

As the first FP-tree constructed to mine association rules, they used a pattern growth method through using patterns of length one in the FP-tree. Every frequent pattern, co-occurring with it in the FP-tree “using the pattern links” are generated and stored in a conditional FP-tree. The mining progression is performed by concatenating the pattern with the ones produced from the conditional FP-tree. The mining process for FP-growth is not like Apriori there is no candidate rule generation, and will not fit into the main memory, when dimensionally large is happen to the mined database.

There are several methods used by Apriori and FP-growth to make comparison on two 10000 record data sets indicates that FP-growth is at least an order of magnitude faster than Apriori since the candidate sets that Apriori must maintain become

extremely large [119]. In addition, the process of searching through the database transactions to update candidate itemsets support counts at any level becomes very expensive for Apriori, especially when the support threshold is set to a small value. As the number of transactions grows, the processing time difference between Apriori and FP-growth becomes larger.

2.6.3 Tid-list Intersection

In order to minimise the number of scans over an input database, the Eclat algorithm has been proposed [48, 65, 120]. This procedure only requires one database scan to address the issue of whether all frequent itemsets can be derived in a single scan. The algorithm that called Eclat is use a vertical database transaction layout, “where frequent itemsets are obtained by applying simple tid-lists intersections”, deprived of complex data structures.

The development of Eclat algorithm variation, called dEclat was proposed by Zaki [49]. The dEclat algorithm uses a new vertical layout representation approach called a “diffset”, which only stores the differences in the transaction identifiers (tids) of a candidate itemset from its generating frequent itemsets. This considerably reduces the size of the memory required to store the tids. Instead of storing the complete tids of each itemset, the diffset approach only stores the difference between the class and its member itemsets. Two itemsets share the same class if they share a common prefix. A class represents items that the prefix can be extended with to obtain a new class. For instance, for a class of itemsets with prefix x , $[x] = \{a_1, a_2, a_3, a_4\}$, we can perform the intersection of xa_i with all xa_j with $j > i$ to get the new classes. From $[x]$, we can obtain classes $[xa_1] = \{a_2, a_3, a_4\}$, $[xa_2] = \{a_3, a_4\}$, $[xa_3] = \{a_4\}$.

Zaki [49] concluded that experimental results on real world data and synthetic data, reveal that dEclat and other vertical techniques like Eclat usually outperform horizontal algorithms like Apriori and FP-growth in relation to processing time and memory usage. Furthermore, dEclat outperforms Eclat on dense data, whereas the size of the data stored by dEclat for sparse databases grows faster than that of Eclat. Consequently, Zaki and Gouda [49], concluded that for dense databases, it is better to start with a diffset representation. Though, it is better to start with a tidlist representation then switch to a diffset at later iterations for sparse databases.

2.7 Pruning Methods in AC

Most of the current rule pruning methods in AC mining are based on database coverage in which they consider a candidate rule is part of the classifier when this rule correctly covers at least one training example during the classification development. Hence, there exist two conditions that must be met before a rule can be inserted into the classifier:

- a) The candidate rule items (left hand side) must be within the items of training examples
- b) The class of the candidate rule (right hand side) must be similar to the training example class

Based on such approach, we argue on two main issues:

- 1) Situations when there is not any candidate rule(s) that covers the training case (no identical similarity). Currently, existing methods use the remaining

unclassified training example to be converted as a default rule. Such an approach may raise higher error rate.

- 2) The condition of having similar classes is unnecessary and can cause overlearning the training data by keeping greedily rules that maximizes the accuracy rate only on the training data without taken into account that the rules are not yet generalized for testing on unseen data which is the main goal of classification in data mining. We believe that by relaxing this constraint and merge it with the partly matching we can end up with a much smaller size of classification model. This can be achieved by allowing the candidate rule to cover more training examples and therefore many redundant lower sorted rules will be unmarked and thus deleted after the building the model step is finished.

However, in the context of data mining, there are some of pruning methods used, which is approved by decision trees, others from statistics such as Estimation, Chi-Square testing, Pessimistic Error. These pruning techniques are in used either during rules discovery phase (Pre-Pruning) such as Pearson's correlation coefficient testing or during the classifier construction phase (Post-pruning) such as Database coverage [43] and Lazy [36]. An early pruning step take place before generating the rules by removing all the rule items which does not passed on the main minsup threshold that might come out in the period to find the frequent ruleitems. This section wills thereby talking about the current pruning methods used by Association classification algorithms.

2.7.1 Database Coverage

The database coverage has been used by CBA [43], CMAR [30] and AACR [73] to choose the subset rules which can make up of classifier. The Database coverage evaluates the complete and total set of rules that are generated against the training of the data set that is basically targeting to keep only the important and complete rules to make up the classifier. Figure 2.1 has shown database reporting explorer method for every rule beginning with the maximum level of the rules, all the training cases that are totally cover up by the rules and the same time the classes are marked for deletion from the training of the dataset with the rules are included even into the classifiers. In a case where the rules are unable to cover particular training case (i.e, when the rule body did not totally match and fit several training cases) after that the rule may be rejected. The method of the database coverage stop once either of the training dataset that it gets is fully covered and it then becomes empty or when there are no specific more rules to be that can be evaluated. In the case when there are no more rules remaining without any evaluation and remain training cases that are not covered are employed to produce the default class rule that usually represents the basically the largest of the frequency class (i.e. the majority class) in the remaining cases that are not classified (unclassified).

It is properly well-known that defaulting class rule is normally used in the process of prediction a step in every cases where there are no classifier rule that is applicable to the trial case. Finally, before database coverage ends, the very first rule that has the lowest number of the errors is usually identified as the main cut-off rule. All the rules that is after this rule are not basically included in the final classifier since they often generate errors[43]. The Database coverage method has been criticized by

Baralis [36] since in many of the cases that it rejected some knowledge that are useful. Otherwise, they suggest that rich classifiers often generate knowledge that are useful and rich in the process of the classification step. Figure 2.2 illustrates the algorithm for the database coverage [43].

```

“Input: The set of sorted rules  $R$  and the training data set  $D$ 
Output: The classifier  $C$ 

For each rule  $r_i$  in  $R$  do
    Mark all applicable cases in  $D$  that fully match  $r_i$ 's body
    If  $r_i$  correctly classifies an case in  $D$ 
    {
        Insert  $r_i$  into  $Cl$ 
        Discard all cases in  $D$  covered by  $r_i$ 
    }
    If  $r_i$  cover no cases in  $D$ 
    {
        Discard  $r_i$ 
    }
}

If  $D$  is not empty
Generate a default rule for the largest frequency class in  $D$  {
Mark the least error rule in  $R$  as a cutoff rule.
}”

```

Figure 2.2: Database Coverage Pruning Methods

2.7.2 Lazy Methods

Lazy Association algorithms [34, 36] is believed that the pruning should also be limited to the rules that is incorrectly covering the training cases in the process of building the classifier. This is because all the rules are generally the ones that resulted from incorrect classification during the prediction of the class label of

testing cases, and therefore they should be the only ones that should be discarded. Database coverage like that of methods used to discard any rule that is unable to be fully covered in a training case and also in class correctness. Otherwise, the Lazy of the Association algorithms store all the rules that is discarded by the database. The like methods is stored in a compact-setfocusing with the aim to make use of them in the process of the prediction step particularly when there is no any primary rules that covers the test case.

Usually, when ranked in the order of descending and total set of rules are established the lazy pruning rule will be applied. For each of the rule that is beginning from the maximum ranked rule, if the chosen rule covers a training case correctly, it will definitely be included into the main primary rule set, and all of its cases that are corresponding will be removed from the training dataset. Whereas, if a rule that is of high ranked is covered correctly the chosen rule training case(s), the chosen rule will be inserted into the main secondary rule set (Spare rule-set). Lastly, if the rule chosen did not cover correctly any of the training case, it will be removed. The method is repeated over and over again until all the rules that discovered are tested or data set training becomes empty. Hence, the result of the lazy pruning will be of two sets of rules, a primary set that retain all the rules that cover in a correct manner of all the training case, also a secondary set that involves the rules that has never been in use during process of the pruning because some rules that of higher rank has covered their training cases.

The distinguished variation between the database coverage and the lazy pruning is that the secondary rules set that are held in the main memory by the lazy method.

However, the classifier generated from CBA based algorithms that uses the database coverage pruning did not contain that of the secondary rules set of basically the lazy pruning, and hence, it is sometimes smaller in the size than that of the lazy based algorithms. In fact, this is an advantage particularly in the area of applications that brings about a concise set of rules in which the end user can control and at the same time maintain.

In an empirical studies, Baralis [36], against the number of UCI data sets has discovered that the use of lazy algorithms reduces more error rate when compared to that of the database coverage. Though, the largest of the classifiers that is derived by the lazy algorithms and also that of the main memory usage limits their main use. Hence, it can be noted that the Lazy based algorithm sometimes scores high in the terms of effectiveness but will low in the efficiency as a result of the large classifier size which will take more time in the generating rules and in the learning the of classifier.

2.7.3 Long Rules Pruning

Cule [121] refers that usually discards a long rules for method called rule evaluation method, which means having a larger confidence values than that of their subset as Li [30] introduced. The evaluation method rule is applied highest confidence value with general rule, which employed so as to prune the particular ones. Furthermore, it delete rules, the rules of redundancy when some of the rules discovered are having a common shared attribute values in their antecedents which often leads a redundant rules and this becomes clear specifically when the size of the classifier becomes large.

Li [30] confirmed that the CMAR is first algorithm that uses the long rules pruning. The long rule pruning will be employed after it ranked the set of the rules based on the confidence, rule length, and support. The CR- tree structure stores the set of the rules and it is necessary that a retrieval query over the tree to be activated in order to check whether the rule can be removed or it can prune any of the existing rules. Chi square testing is used in each $R: r_i \rightarrow c$, in order to determine whether the r_i is positively correlated with c or not. The algorithm only chooses rules that positively form the classifier. There are some AC methods that use this particular type of pruning, including ARC-BC [122], and the Negative Rules [44]. Experimental results reported in [30] found that using the pruning method will be positively affect the effectiveness when trying to contrasted with the other methods.

2.7.4 Mathematical Based Pruning

Some mathematical-based pruning methods have been proposed for the classification and AC. Most of them usually tend to measure the correlation between different object so as to decide whether they are correlated or not in order to make a decision either to prune a rule or considering it in the classifier. Here a number of pruning methods will be discussed.

2.7.4.1 Chi-Square Testing

The chi-square test (χ^2) proposed by [90] is normally applied to decide whether there is a significant difference between the observed frequencies and the expected frequencies in one or more categories. It is defined as a known discrete data hypothesis in mathematics that examines the relationship between two objects in

order to decide whether they are correlated or not [123]. The evaluation using χ^2 to decide the independencies or the correlation of a group of objects is given as:

$$a. \chi^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i} \quad (2.20)$$

b. Where O_i is the observed frequency and E_i is the expected frequency.

If they are especially different from the frequencies expected and the frequencies observed, the statement that they may be related is dropped.

CMAR is the first algorithm of AC which uses a weighted version of χ^2 . It evaluates the correlation between the antecedent and the consequent of the rule and thereby removes rules which are negatively correlated. A rule $R: Antecedent \rightarrow c$ is removed if that of the class c is not positively correlated with that of the antecedent. Alternatively, if the output of the correlation is more than a certain threshold, this shows a positive correlation. Otherwise, R will be discarded due to negative correlation that exists in R . To clarify, for R , assume $Support(c)$ indicates the number of training instances that are associated with the class c and $Support(Antecedent)$ also indicate the number of training cases associated with the R 's antecedent. Also assume that $|T|$ denote the size of the training data set. The weighted chi-square denoted $Max \chi^2$ of R is defined as:

$$\chi^2 = (\min\{Support(Antecedent), Support(c)\} - \frac{Support(Antecedent)Support(c)}{|T|})^2 |T| u \quad (2.5)$$

Where,

$$u = \frac{1}{\text{Support}(\text{Antecedent})\text{Support}(c)} + \frac{1}{\text{Support}(\text{Antecedent})(|T| - \text{Support}(c))} + \frac{1}{(|T| - \text{Support}(\text{Antecedent}))\text{Support}(c)} + \frac{1}{(|T| - \text{Support}(\text{Antecedent}))(|T| - \text{Support}(c))}$$

Statistical Association Rule Classification (SARC) is an AC algorithm [124] used chi-square in the rule pruning step, any potential rule that are negatively correlated according to chi-square gets deleted from the discovered rule set. The rule significance test is performed after the rule has passed the confidence and support tests.

2.7.4.2 Pessimistic Error Estimation

Pessimistic error estimation is mainly used in data mining within decision trees [41] in order to decide whether to replace a sub-tree with a leaf node or to keep the sub-tree unchanged. The method of replacing a sub-tree with a leaf is called “sub-tree replacement”, and the error rate is computed using the pessimistic measure on the training dataset. To clarify, the probability of an error at a node v is giving by the following relation:

$$q(v) = \frac{N_v - N_{v,c} + 0.5}{N_v} \quad (2.21)$$

Where N_v denotes the number of training cases at node v is, $N_{v,c}$ is the number of training cases belonging to the largest frequency class at node v .

The error rate at sub-tree ST as follow,

$$q(T) = \frac{\sum_{l \in \text{leaves}(T)} N_l - N_{l,c} + 0.5}{\sum_{l \in \text{leaves}(T)} N_l}. \quad (2.22)$$

Where, the sub-tree ST is pruned if $q(v) \leq q(ST)$.

The pessimistic error estimation has been exploited successfully in the decision tree algorithms which include C4.5 and C5.0 [125]. In AC mining, the first of the algorithm which had used the pessimistic error pruning is the CBA. For a rule R, CBA removes one of that its attribute value also its antecedent so to make a new rule R', it then compares the estimated error of R' with that of R. If the expected error of R' is smaller than that of R, then the original rule R is replaced with that of the new rule R'.

It must be noted here that the CBA uses two pruning methods, which is the pessimistic error and also the database coverage. Some studies found that by employing several pruning procedures which may affect the accuracy rate [36, 126].

2.7.4.3 Pearson's Correlation Coefficient Testing

The other statistical-based approaches that can be used to measure the strength of the correlation between two particular objects. HMAC [127] is seen as one of the Association classification approaches that employed this measure. After producing the set of CARs, HMAC that uses two pruning procedures namely (1) Pearson's correlation coefficient procedure and that of (2) redundant rule, ranks the rules based on the procedure of ranking [127]. HMAC begins with the Pearson's Correlation Coefficient and applies it for every positive class rule RPC in order to measure the

correlation strength between the antecedent, consequent and the class label rules of the item.

Although it is revealed in the experimental results [127] that algorithms using Pearson's test can result in gaining good accuracy results, it is difficult to validate this as insufficient experimental results are available and much of the information relating to their generation is absent.

2.7.5 Laplace Accuracy

Clark [128] confirmed that the post-pruning method is Laplace accuracy, are invoked in the process of the construction of the classifier, it is used in other to estimate the error normal ratio of a rule $r: p_1p_2\dots p_n \rightarrow c$, the accuracy expected for a particular given rule r is computed through the following formula:

$$\text{Laplace}(r) = \frac{(p_c(r) + 1)}{(p_{tot}(r) + m)} \quad (2.22)$$

The identification of the formula is " $P_c(r)$ which denotes the number of the training cases that is covered by r with class c . $P_{tot}(r)$ is the number of the training cases that is matches r 's condition and m is the number of class that has been labels in the domain". In the Association classification is adopted a Laplace in recent CPAR algorithm [35].

Hun [35] presented a report for results of experimental against 26 datasets from UCI repository showed that CPAR. The outcome that used Laplace accuracy algorithm is

bad in classification accuracy rate but better in efficiency when compared to CMAR and CBA.

2.7.6 Redundant Rule Pruning

Rules body is a concept that refers to the all of the attribute value combinations derived from Association classification approaches. Therefore, some training items in classifier bodies may share with rules that are used in their build. Finally, many general rules may include some specific rules. The serious problem, such the huge of the number of the generated rules in AC approach.

Li [30] proposed a new rule pruning method that is named redundant rule pruning method. This method discards specific rules that have a confidence value less than general rule the method is working as follows: once the set of rules being generated and sorted, redundant rule evaluation is invoked to discard all rules such as $I' \rightarrow c$ from the set of generated rules where there are some general rules such as $I \rightarrow c$ with higher rank and $I \subseteq I'$. This method is notably reduces the size of the classifier since it reduce the rules redundancy. Redundant rule pruning method have been used in several Association classification algorithms including CMAR [30], ARC-BC [129], ACN [45], CACA [38].

2.7.7 Conflicting Rules

In some datasets in which they considered dense datasets or multi-label where multiple class labels associated with a training case, there is a possibility to have two rules with same rule body but associated with two different class labels, such as the following two rules: $x \rightarrow ca$ and $x \rightarrow cb$, conflicting rules pruning method [130]

consider such rules conflicting, discards them and prevent them to take any role in the classifier. Nevertheless, Thabtah [46] has a study presented the experimental results in MMAC algorithm which rules could appear useful knowledge subsequently they confidence requirements and pass support.

2.7.8 Compact Rule Set

Tang [38] states that the CACA is the Association classification algorithm, which combines the two stages into one stage. The stages are classifier learning and Rules generation. In this method, an Order Rule Tree structure (OR-Tree) is designed to store and rank the set of generated rules; after generating the set of rules that satisfies the MinSupp and MinConf thresholds such as $r_i (a_{i1}, a_{i2} \dots a_{in}, c_i)$, a redundant pruning procedure is applied on the set of rules R, CACA consider a rule r_i redundant if one of four is met, for a given rules r_1 and r_2 , r_2 , r_3 considered redundant if:

$$1) r_1 = \langle I_1, c_i \rangle \text{ and } r_2 = \langle I_2, c_j \rangle \text{ but } r_1 > r_2$$

$$2) r_1 = \langle I_1, c_i \rangle \text{ and } r_2 = \langle I_2, c_j \rangle \text{ but } I_1 \subset I_2, \text{ and } r_1 > r_2$$

i.e.(rules have different class label)

$$3) r_1 = \langle I_1, c_i \rangle \text{ and } r_2 = \langle I_2, c_i \rangle \text{ but } I_1 \subset I_2, \text{ and } r_1 > r_2$$

i.e. (both rules have the same class label)

$$4) r_1 = \langle I_1, c_i \rangle \text{ and } r_2 = \langle I_2, c_i \rangle \text{ but } I_1 \subset I_2, \text{ and } r_2 > r_1. \text{ For } r_3 = \langle I_3, c_j \rangle, I_1 \subset I_3, r_3, r_2, r_1.$$

For more illustration, the compact rules set \rightarrow is extracted, \mathfrak{R} is the set of the original rules R excluding all redundant rules i.e. $\mathfrak{R} = R - \text{Redundant rules}$ which ensures that all redundant rules are not taking any role in the classifier.

That means, $r \in \mathfrak{R}$, the attribute values $(a_{i1}, a_{i2} \dots a_{in})$ will be stored as nodes in the OR-Tree in descending order (The most important rule is stored in the closest node

to the root while the one with less importance in the node after and so on) according to their frequency in the training dataset D whereas other details such as class label, support and confidence are stored in the last node in the leaf. The proposed pruning procedure here is works as follows: However, $r_i = I_x \rightarrow c_1$ if

$$"supp(r_i) / conf(r_i) \times (1 - conf(r_i)) < minsupp" \quad (2.23)$$

Then stop mining $r_i = I_y \rightarrow c_j$ where $I_y \supseteq I_x$. Hence, r_i will be discarded.

Based on the results reported by Tang [38] this Algorithm that uses compact rule set can competitively classify a bit better in efficiency and effectiveness than other AC algorithms such as CBA and MCAR. In terms of time taking in mining the rules, that will be remarkably hence enhanced efficiency and reduced due to the cut in the items.

2.7.9 I-Prune

Baralis [53] proposed Item prune as a pre-pruning method that tends to mark uninteresting items based on interestingness measure (correlation measures e.g “Chi Square”, “Lift”, “Odd ration”) and remove them and use only interesting items to build a high quality rules which will be used in building the classification model. This approach reduces the number of generated rule through pruning step, in addition the time taken for learning the classifier.

Several AC algorithms such as CBA [43], CPAR [35], CMAR [30], MCAR [37] consider an item interesting according to the support count. reflect an item interesting according to the support count. Otherwise, I-prune chooses only those are correlated and frequent. Assume a class c is correlated to item i, an interestingness

measure is given as follows: if interestedness-measure $(i,c) >$ predefined-threshold then i is selected else items are discarded as soon as detected. Assume I is a subset of frequent and correlated items with respect to class c , set of rules R is generated for c .; only the rules that contains interesting items are generated. However, the I-prune method discards some useful classification rule. Baralis [53] states that the Chi Square is the best correlation measure based on experimental results for the set of all measures used, which mean respect to effectiveness.

2.7.10 PCBA-based Pruning

Chen [131] presented a PCBA pruning method as new pruning method, which consider class unbalancing. The purpose of this method is to attempt to deal with imbalanced class that will happen when applied to the Association classification. Moreover, one fixed minsupp is used in AC algorithms as well as minconf. However, this approach works well when balanced data is used. Otherwise, “under-sampling” is a concept to distribution the rule of each class by uses minconf values and different minsupp through this algorithm.

2.8 The Methods of Prediction

Assigning the appropriate class labels to test cases is the last step in the life cycle of any classification algorithm. This exercise can called as class assignment or class prediction. Actually, AC mining has a number of different approaches for class assignment task, the highest ranked rule in the classifier are adopted from several methods, as well as some methods with single rule prediction [38, 43, 120] and other. So, different prediction methods are review to understand the main characteristics and their employed.

2.8.1 Single Rule Class Assignment

Liu [43] presents the CBA algorithm that is illustrated in Figure 2.3 as a basic idea of one rule prediction. The steps of this method start with the classifier which is constructed. After it builds the classifier, the rules within it are sorted in descending order according to support thresholds and confidence. On the other hand, “a test case is about to be forecast, CBA iterates over the rules in the classifier and assigns the class associated with the highest sorted rule that matches the test case body to the test case. In cases there are no rules matches the test case body, CBA takes on the default class and assigns it to the test case. After the dissemination of CBA algorithm, a number of other AC algorithms have employed the one rule prediction method such as in” [38, 45, 112, 118, 132, 133].

“Input: Classifier (R), test dataset (Ts), array Tr Output: error rate Pe ”
Given a test data (T), the classification process works as follow: 1 \forall test case ts in Ts Do 2 \forall rule r in the set of ranked rules R Do 3 Find all applicable rules that match ts body and store them in Tr 4 If Tr is not empty Do 5 If there exists a rule r that fully matches ts condition 6 assign r 's class to ts 7 } 8 else assign the default class to ts 9 } 10 empty Tr 11 } 12 } 13 calculate the total number of errors of Ts .”

Figure 2.3: Single Rule Class Assignment Methods

2.8.2 Class Assignment Based on Group of Rules

The performance of single rule prediction method is well specifically when there is just a single rule applicable to a test case. On the other hand, the single rule prediction method has been questionable where close confidence values with more than one rule are applicable to the test case. Therefore, it is an inappropriate where selection of a single rule is used to make the class assignment, because of using all rules contributing to the prediction decision. Consequently, different multiple rules class assignment methods are discussed in the following section.

2.8.2.1 Weighted Chi-Square Method

Li [30] states that the first algorithm of AC is CMAR which employed weighted Chi-Square ($\text{Max } \chi^2$) for class assignment task. In est cases, all applicable rules will choses when CMAR is applied and then assesses their correlations. The correlation measures the strength of the rules based on the support and class frequency in the testing data set.

The algorithm selects the set of the ranked rules R in the classifier, the subset of rules, R_k that may satisfies test case condition. If all rules in R_k have the identical class, then that class will be assigned to t_s . However, if the rules in R_k associate with different classes, CMAR divides them into groups based on the classes and computes the strength of each group. The group's strength is identified by different parameters such as the support and correlation between the rules in a group. i.e. ($\text{Max } \chi^2$). Finally, to the test case t_s , the CMAR algorithm references the class of the largest group strength.

Thus, rule R as illustrated: $Cond \rightarrow c$, “assume Support (Cond) represents the number of training cases associated with rule body Cond and Support(c) denotes the number of training cases associated with class c. Also assume that $|T|$ represents the training dataset size”. The definition of Max (χ^2) of Rk is:

$$Max \chi^2 = (\min\{Support(Cond), Support(c)\} - \frac{Support(Cond)Support(c)}{|T|})^2 |T| u \quad (2.24)$$

Where,

$$u = \frac{1}{Support(Cond)Support(c)} + \frac{1}{Support(Cond)(|T| - Support(c))} + \frac{1}{(|T| - Support(Cond))Support(c)} + \frac{1}{(|T| - Support(Cond))(|T| - Support(c))}$$

AC algorithms have adopted Max (χ^2) to the class assignment task after being introduced by CMAR. Furthermore, [134] used a closely similar class assignment method of CMAR, where the class of the subset of rules in R_s with the dominant class gets assigned to the test case T_s .

The experimental results reported in [30] showed that classification procedures that employ a group of correlated rules for prediction slightly improve the classification rate when contrasted to other methods.

2.8.2.2 Laplace based Method

CPAR algorithm is the first AC learning technique that used “Laplace Accuracy” to evaluate the rules and assign the class labels to the test cases during class assignment step. Once all rules are found, ranked and the classifier constructed, and a test case (ts) is about to be predicted, CPAR goes over the rule set and marks all rules in the

classifier that may cover ts . If more than one rule is applicable to ts , CPAR divides them into groups according to the classes, and calculates the average expected accuracy for each group. Lastly, the class with the largest average expected accuracy value is assigned to ts . The computed for the expected accuracy of each a rule (R) is as follows:

$$\text{Laplace}(R) = \frac{(p_c(R) + 1)}{(p_{tot}(R) + p)} \quad (2.25)$$

Where,

p is the number of classes in the training data set

$p_{tot}(R)$ is the number of cases matching r antecedent

$p_c(R)$ is the number of training cases covered by R that belong to class c .

The CPAR algorithm will used successfully the Laplace accuracy [35] the largest rule has positively affect the classification accuracy that will happen to ensure about the accuracy contribute in class assignment for test cases.

2.8.2.3 Dominant Class and Highest Confidence Method

Two closely similar prediction methods that use multiple rules to predict the class labels for test cases were proposed in [135]. The first method is called “Partial Dominant Class”, which marks all rules in the classifier that are applicable (Partially match the test case body) to the test case, and then groups them according to class labels, and assigns the test case the class of the group which contains the largest number of rules applicable to that case. In cases where no rules are applicable to the test case, the default class (Majority class) will be assigned to that case.

The second prediction method is called “Highest Group Confidence”, which works similar to the “Partial Dominant Class” method in the way of marking and dividing the applicable rules into groups based on the classes. However, the “Highest Group Confidence” computes the average confidence value for each group and assigns the class of the highest average group confidence to the test case. In cases where no rule matches the test case, the default class will be assigned to that case.

2.8.3 Predictive Confidence

In class assignment step, the foremost weight considered for rule in selecting the right rule to fire for class assignment of test cases is the confidence value. However, Do [136] states that to discriminate among rules in the classifier, which means the confidence that is calculated from the training data for rules is not enough. Hence, besides the confidence value there should be other criteria for rule choice in prediction. For example, the “predictive confidence” measure that can be measured for each rule in the classifier and from the test data set.

The predictive confidence criterion represents the average classification accuracy for a rule r when assigning classes to test data case. Given for more clear a rule ri : $ListOfItems \rightarrow c$, assume that there is “A” parameter which represents the test cases that matches ri condition and belonging to class label c , and a “B” parameter which represents the test cases matching only ri condition. Currently, “when ri is applied on the test data set, ri will correctly predict “A” test cases with prediction accuracy of (A/B) which is simply the confidence value of (ri) on the test data set. This is simply the definition of the prediction accuracy of the rule that has been implemented on a recent AC algorithm called AC-S” [136]. From the training data

set, the AC-S algorithm is employed to choose the right rules for prediction instead of the confidence value computed. Do [136] confirms that the AC-S algorithm is very competitive to common AC algorithms such as CMAR, and CBA.

2.9 Comparison of Association Classification Algorithm

Figure 2.4 summarizes algorithms discussed in this chapter. All of the algorithms require rule discovery, rule pruning and rule prediction methods. The output of a rule discovery is frequent item based on a set of minimum support and minimum confident values and a set of association rules. The rule pruning method matches the training data and the obtained association rules where any rule that matches with at least one training data will be feed into the classifier, while the rules, that are not, will be deleted. This is followed by the rule prediction method that test the classifier.

One of the rule discovery methods is the Apriori which is used in the CBA [43], 2-PS[117], CAN[119] AND PCBA[64]. It uses horizontal data layout to represent the data, this means that it needs multi scan to get the frequent item, hence, it leads to increase the computational time. In addition, Apriori relies only on the single minimum support value to generate the rule and this increases the number of rules especially when the minimum support is high. On the other hand, the FP growth which is another method in rule discovery needs to scan the intersection between items in the training data to get the frequent item. This will also increase the computational time. Another method for rule discovery is the TID-list intersection, and it uses vertical representation. It requires one time scan to get the frequent item, but the frequent item set is only based on data and may not be sufficient to discover the rules from vertical data in order to produce relevant frequent rule item.

The pruning algorithm is used to match between the training dataset and the generated rules. The Database coverage applied on CBA [43], MCAR [37], MMCA, 2-SP, CMAR [30], ACCF and AACR [73] evaluates the complete part of the rules against the training of the data set. Reach a classified data requires a Full match between the rule and the training data set. The similarity of the class is important between the training data set and rule. Any rule is not fully match with similar class will get the default class (default class the class have exist in the dataset). Boosting association pruning method applied on CPAR [35], ICPAR[36] has full match between the rule and training data set with class similarity, lazy pruning divide two sets of rules, a primary set that retain all the rules that cover in a correct manner of all the training case, also a secondary set that involves the rules that has never been in use during process of the pruning because some rules of higher rank has covered their training cases. Lazy pruning keep the secondary rules set that are held in the main memory by the lazy method. Another algorithm for pruning is long rule pruning applied highest minimum confident value to match the rule with training cases, and delete rules with small confident. This will delete the good rules and will decrease the accuracy in some cases.

The prediction algorithm is used to make a match between the testing data set and the generated rule, and the outcome is the classification accuracy. There are mainly four types of prediction which are; Single rule prediction, CMAR multiple label, CPAR multi label and the Dominant factor. The first algorithm single rule, is applied in CBA [43], MCAR [37], MMCA [46], CNA[44], ACCR[], ACCF[112], and CACA [38]. It uses the highest sorted rule (one rule only depend on minimum

support and minimum confident) in the classifier and assigns the class with the sorted rule to match the test case, if the rule does not match a test case body, it take the default class. The CMAR multiple label prediction algorithm is applied in CMAR [30], where divide the rules to groups depend on the support and class frequency. The multi-label prediction algorithm applied in CPAR [35] divides the rules into groups depending on the class. The class with the largest average expected accuracy will be used to test a given case. Such an approach may reduce the accuracy as there may be cases that do not have same set of frequent itemset like in majority of the cases.



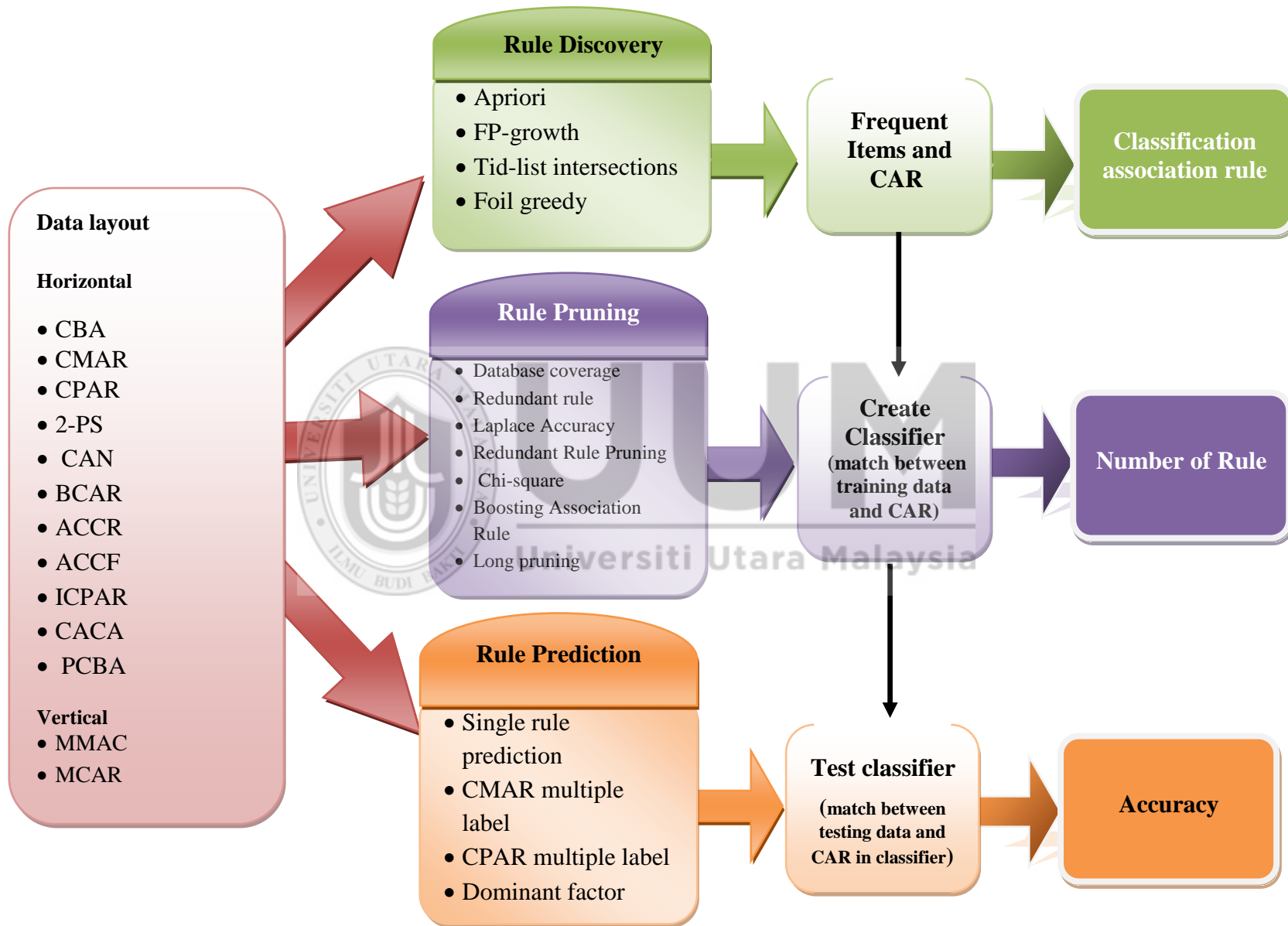


Figure 2 4: Theoretical Framework

The different approach used in each algorithm in data layout, discovering rules, rules ranking, rule pruning and predication is tabulated in Table 2.3.

Table 2.3
Summary of AC algorithms

Name	Data Layout	Rule Discovery	Ranking	Pruning	Prediction Method
CBA [43]	Horizontal	Apriori candidate generation	Support, confidence, rules generated first	Pessimistic error, database coverage	single rule prediction
CMAR [113]	Horizontal	FP-growth approach	Support, confidence, rules cardinality	Chi-square, database coverage, redundant rule	CMAR multiple label
CPAR [65]	Horizontal	Foil greedy	Support, confidence, rules cardinality	Laplacee expected error estimate	CPAR multiple label
MMAC [57]	Vertical	Tid-list intersections and recursive learning	Support, confidence, cardinality, class distribution frequency	Database coverage	single rule prediction
MCAR [111]	Vertical	Tid-list intersections	Support, confidence, cardinality, class distribution frequency	Database coverage	single rule prediction
2-PS [117]	Horizontal	Apriori candidate generation	Support, confidence, rules cardinality	Database coverage	dominant factor
ACN [119]	Horizontal	Apriori candidate generation + Negative Rules	Confidence, rules Correlations, Support, rules cardinality , Positive Rules	redundant rule , pearson's correlation coefficient	single rule prediction
BCAR [106]	Horizontal	Boosting Association Rule	Support, confidence, cardinality	Boosting Weak Association Rule	normalized prediction score model
ACCR [123]	Horizontal	Cluster-based association rule	Support, confidence, cardinality	Pessimistic error, database coverage	single rule prediction
ACCF [120]	Vertical	Charm	Support, confidence, rules generated first	Pessimistic error, database coverage	single rule prediction
CACA [118]	Vertical	Class-based Association classification	Support, confidence, rules generated first	Compact set, redundant rule	single rule prediction
ICPAR [135]	Horizontal	Foil greedy	Support, confidence, cardinality	Laplace Accuracy	Multi-Label-ICPAR
PCBA [64]	Horizontal	Apriori candidate generation	Support, confidence, cardinality	PCBA pruning	SPA probabilistic

2.10 Chapter Summary

This chapter presents the literature in text mining, particularly the ones in Associative Classification. The AC is a data mining approach which builds more accurate classifiers than traditional classification approaches such as decision trees and rule induction. By integrating association rule mining with classification, AC has two main phases which are rule generation and classifier development. A number of well-known AC techniques have been presented in this chapter. The literature addresses the methods used in rule discovery, rule pruning and class prediction method. Most of existing techniques employ the Tid-list intersections with either horizontal or vertical layout of data representation. Rule discovery is performed on the produced layout. This is followed by using various pruning methods to determine most relevant rules (i.e rule pruning). Rule pruning methods focus on matching both sides of rules (i.e the left-hand side and right-hand side). Once a set of rules is obtained, it will be used for class prediction of a new dataset. And this prediction can be done using several methods such as single rule or considering dominant factor.

This thesis proposes the combination of vertical and horizontal layout to be used by Tid-list intersection for rule discovery. This is followed by introducing a partly rule match method in determining relevant rules. To improve the classification accuracy, this study proposes the use of group-based prediction method in determining the best class for a given test case. In the upcoming chapter (i.e Chapter 3), relevant steps to achieve the stated objectives are presented.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

In this chapter, the methodology used for this research is presented. More specifically, the research is of an experimental since the performance of different text mining algorithms particularly Association and other rule-based are investigated on a collection of text documents. The proposed model in Figure 3.1 follows the three main stages understanding in this study: Pre-processing; Design Classifier model; and Develop classifier model and Evaluation.

In the process of data collection the data were collected for two experiments. For the first experiment data set was the use of fourteen UCI [76] data and for second experiment the most populated categories of the Reuters-21578 [69] test collection were used. In pre-processing, the number of pre-processing methods includes feature selection and vector representation that are applied in order to reduce error rate because unprocessed data contain sparse, unstructured patterns, noise such as records redundancy, incomplete transactions and missing values. In designing the classifier model, Rule discovery, Rule ranking, Rule pruning and Rule prediction methods were employed in order to get a better predictor classifier model. In the development of classifier model the proposed AC algorithm was implemented using Visual Basic VB, and the comparative study will be based on our implementation as well as data mining packages called WEKA [137] and CBA [43] , with justification for using this model were also given in this section.

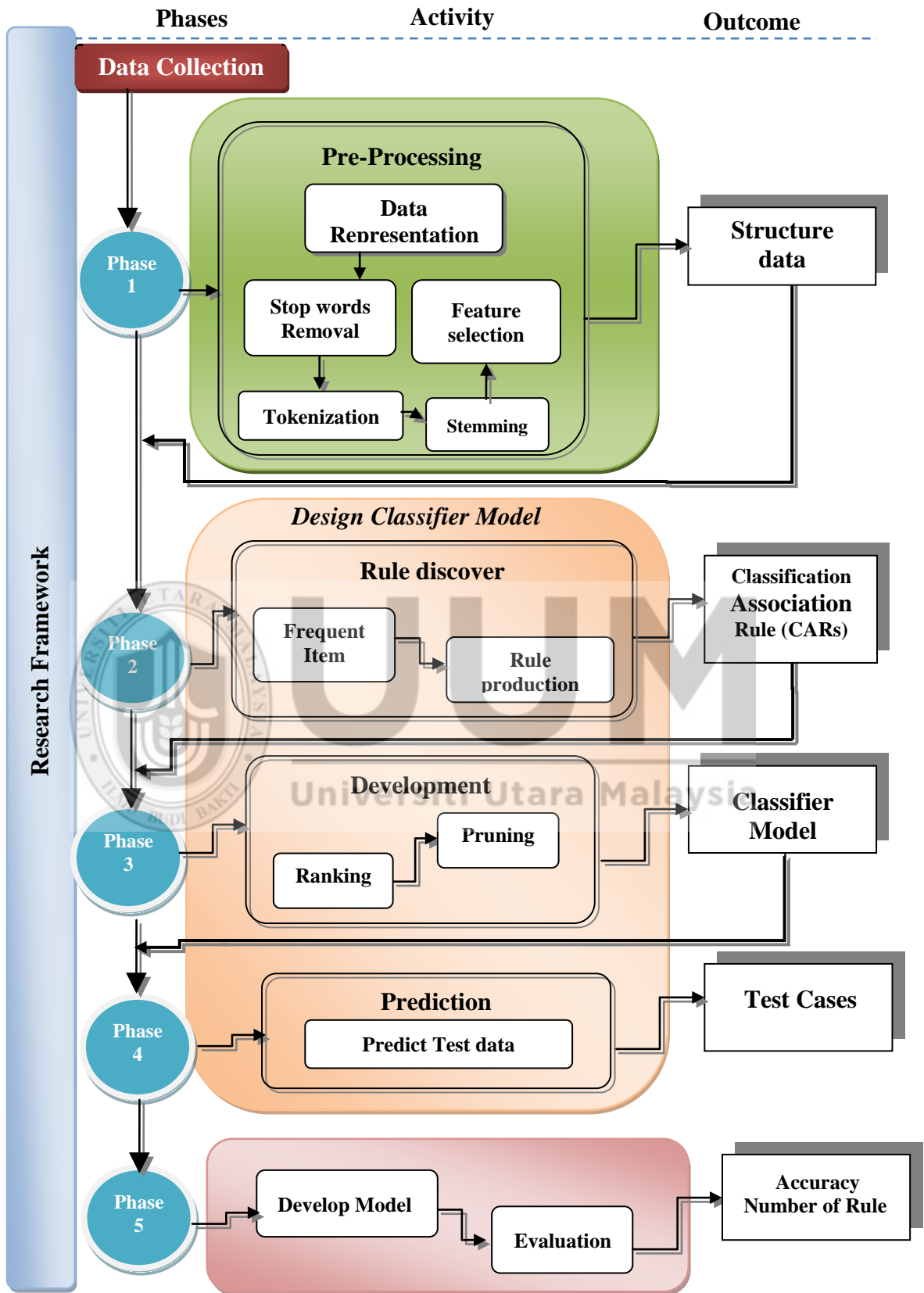


Figure 3.1: Research Methodology

3.2 Data Collection

In this research there is a two datasets was employed. The first uses fifteen (15) UCI dataset. UCI data sets are the most widely used benchmark for empirical evaluation of new and existing learning algorithms [138]. On the other hand, the second experiment is performed on seven (7) most populated categories of the Reuters-21578 collection [62]. General description of UCI dataset is displayed in Table 3.1. The datasets are of different sizes, ranging from 14 to 8124. These datasets were divided into three; small, medium and large. Datasets with less than 200 instances are group as small while the ones with larger than 600 are considered as large. Hence, datasets that contain instance between 201 and 599 are categorized as medium size dataset. Data in Table 3.1 also depicts the type of data in the dataset. This includes Nominal, Numeric, Boolean and Categorical [76].

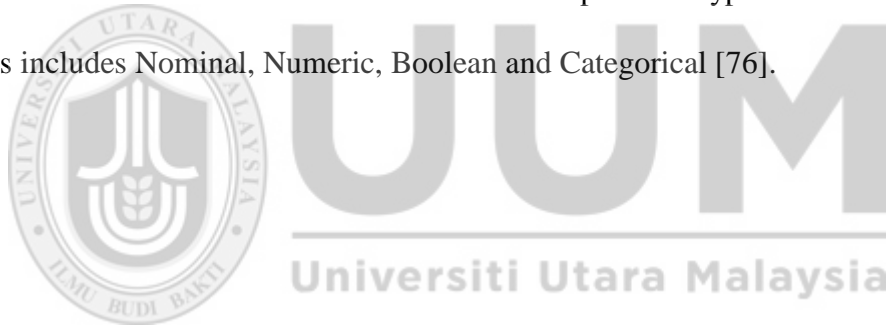


Table 3.1

Description of UCI Data Sets

Data set	Instances Size	Number of Class	Data type	Data Size
Weather	14	2	Nominal	small
Labor	57	2	Numeric, Boolean	small
Lymph	148	4	Numeric	small
Iris	150	3	Numeric	small
Wined	178	3	Numeric	small
Glass	214	7	Numeric	medium
Heart-s	294	2	Real, Binary ,Nominal	medium
Cleve	303	2	Nominal, Numeric	medium
Vote	435	2	Boolean	medium
Balance-scale	625	3	Numeric	medium
Austra	690	2	Numeric, Categorical	Large
Breast	699	2	Numeric	Large
Pima	768	2	Numeric	Large
Led7	3200	10	Numeric	Large
Mushroom	8124	2	Nominal	Large

Based on the previous studies literatures [2,86,105] in text mining, the most commonly utilized data set is the Reuters-21578. Documents in the Reuters-21578 collection one of the appear on the Reuters newswire and were indexed by personnel. This study requires Reuters-21578 version ModApte which comprise of 9,174 documents. The data divided by expert into 2,579 of testing and 6,630 training documents. Table 3.2 shows the number of documents in training and testing sets per category REUTERS-21578. Sample of the dataset is provided in Appendix A.

Table 3.2

Number of Training and Testing Document in REUTERS-21578

Category	Training	Testing
Acq	1650	719
Crude	389	189
Earn	2877	1078
Grain	433	149
Interest	347	130
Money-FX	538	197
Trade	396	117

3.3 Data Pre-processing

One of the most important stages in text classification is the preparation of the input data. The textual data sets are and may contain noise, unstructured, and often sparse, such as missing values, incomplete transactions, record redundancy ,etc. [71,69,139]. Hence, the quality of the request may be affected by the high quality of input data. Figure 3.3 displays the utilised pre-processing methods including feature selection, vector representation. Here are the different steps in pre-processing phase according to [70,139].

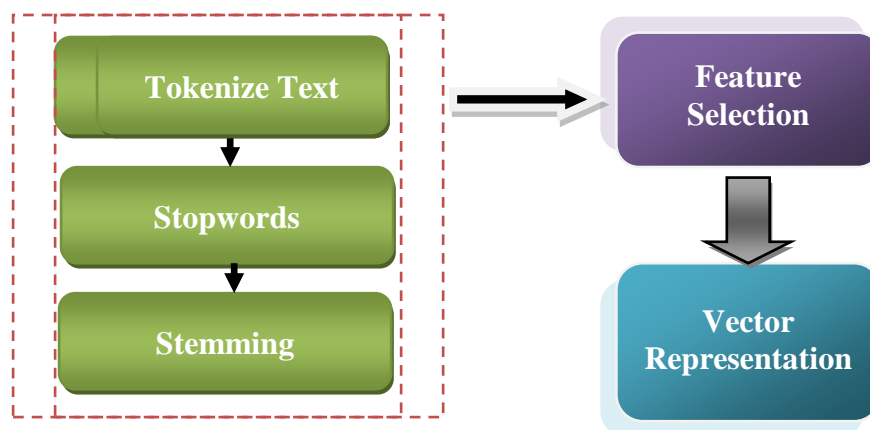


Figure 3.2: Pre-processing Operation in Text Mining

3.3.1 Tokenisation

Feldman [140] defined the Tokenization method as a process to improve meaningful tokens through breaking up the sequence character, which means the text document are broken into sentences, and words. Furthermore, in the Explorer GUI will used WEKA filtering to tokenise the input document. Example from “Reuters earn dataset” the implement of Tokenisation method is illustrated below:

Its board of director's approved  **board of directors approved**

3.3.2 Stopwords Removal

Often text documents contain numerous words that are meaningless for the learning algorithms such as “before”, “after”, "the", "of", “in”, "on", “out”, etc. These words should be deleted during the preprocessing phase, since such words negatively impact the resulting classifier [141]. In the proposed model, Google stopwords list will be employed on the Reuters textual collection. Google stopwords list “appendix B” will be employed on the Reuters textual collection. Example from “Reuters earn dataset” the implement of Stopwords removal method is illustrated below:

Its board of director's approved  **Board directors approved**

3.3.3 Stemming

The stemming is the process of converting words into their root, for instance. playing to play, construction to construct, diver to dive [140]. We use WEKA stemmer [64] on the Reuter data collection. Example from “Reuters earn dataset” the implement of Stemming removal method is illustrated below:

Its board of director's approved  **board director approve**

3.3.4 Data Representation

In the proposed model we use a data format based on combining vertical and horizontal data layouts to represent the data. To the best of our knowledge there is no AC technique that utilizes integration of vertical and horizontal data layouts for data representation. An example of a vertical data format is MCAR [37] algorithm which employs a tid-list data structure to hold the appearances of the item in the input data set. Our model differs from MCAR data layout in the way of representing each item. In our model, an item is represented by the line number of which the first item occurs in the data set as well as the column number of that item. Meaning each item is converted into ColumnId, RowId representation which are simple integers and therefore the search for items to compute the support and confidence values during the rule discovery process requires less time memory. On the other hand, MCAR algorithm uses two data structures to represent the input data set; one contains the tid-list of each item and one for the occurrences of the class labels. Table 3.3 shows an example of input dataset, while Table 3.4 depicts the proposed representation.

*Table 3.3
Examples of Item Found on Each Line*

TID	Items
1	sea,port , wind
2	port,aqaba
3	port,corn
4	sea,port, aqaba
5	sea,corn
6	port, corn
7	sea, corn
8	sea,port, corn,wind
9	sea,port, corn

Table 3.4
Representation of Item

TID	Item Ids
1	(1)1,(2)1,(3)1
2	(2)1,(4)2
3	(2)1,(5)3
4	(1)1,(2)1,(4)2
5	(1)1,(5)3
6	(2)1,(5)3
7	(1)1,(5)3
8	(1)1,(2)1,(5)3,(3)1
9	(1)1,(2)1,(5)3

As illustrated above, for each transaction a unique integer value will be added as a minimum initialization done for data. The TID in Table 3.4 refers to the transaction ID. The item id is referring to its integer representation, which means it is replaced for each Item with integer values of two parts. Row Id Ex and column Ids, (column1 “sea”, row 1 it represent ((1)1). Now, the algorithm will computing the support of an item to determine whether it is frequent or not, an aggregation function gets invoked to group each item and count their appearances within the data structure. This process is straight-forward and can be implemented for support and confidence calculations which make the process of determining the rules straightforward. In Section 4.3 will be show the usefulness of the data representation in the proposed AC method within the rule production step.

3.3.5 Feature Selection

In our model we use a combination of vector representation and term frequency to convert the high dimensionality of the collection of Reuter text into a matrix. Moreover, the model will compute the frequent items by using simple TID list intersections.

Term Frequency (TF) is employed to measure the significance of the keyword and their contribution to the output classifier [142]. Specifically, TF is one of the term weighting methods that measures the frequency of the keyword in document and is given in the equation below:

$$tf(f_j, d_j) = \frac{freq_{ij}}{\max_k freq_{kj}} \quad 3.1$$

However, the text data shifts into a classical data mining encoding, that will be as intermediate point, this step depend on converting the text into a standard numerical form which is suitable for algorithms of learning (structured data), therefore, the features was chosen by using TF. [143].

3.4 Design Classifier Model

The four main steps in the design classifier phase will be discussed in the next subsection, and is illustrated in Figure 3.1.

3.4.1 Rule Discovery

In this section, we briefly explain how support and confidence for rule items are calculated using an example and show how rules are generated. Association rule discovery contains two stages namely, frequent itemset discovery and confident rule

foundation [25]. In this study, the frequent items use an intersection method based on the Tid-list [48] to compute the support and confidence values of ruleitems having size greater than one. For instance, for a class of itemsets with prefix x , as follows the formula

$$[x] = \{a_1, a_2, a_3, a_4\}$$

The intersection of xa_i will perform with all xa_j with $j>i$ to get the new classes. From $[x]$, we can obtain classes as follows the formula

$$[xa_1] = \{a_2, a_3, a_4\}, [xa_2] = \{a_3, a_4\}, [xa_3] = \{a_4\}$$

Thus, after all frequent ruleitems are identified the confident rules, which be the second stage. The confident rule for each itemset of them that passes the minconf threshold, a single rule is generated of the form: $X \rightarrow C$, where C is the largest frequency class associated with itemset X in the training data.

The frequent one-item is counted only once by the training data set that come from the rule production, as well as, discovers those that passes the MinSupp. The data structure stores the frequent one- items in a vertical format, after the items were scanned and determined. However, the items will be removed if they did not pass the MinSupp. Next, the candidate two-item is produced by using the Tid-lists of the frequent one-item, simply by intersecting the Tid-lists of any two disjoint one-items. Furthermore, the confidence value of Class Association Rule (CAR) is larger than the MinConf threshold which is validated by the AC algorithm; otherwise, if the rule item is deleted, that means the CARs represents items are statistically representative and have high confidence values. In section 4.3 will describe the entire proposed rule discovery.

3.4.2 Rule Ranking

The most important step in AC is a rule ranking which helps to choose the most effective rules for prediction. However, in the process of building the classifier will be sorting on the rules that perform through AC technique. Moreover, the first step toward removing useless rules and pruning noise is sorting the rules. Actually, the rules must be arranged to give the higher quality rule a better priority, which will help to build up the classifier and prior to prune redundant rules.

Furthermore, the technique of rule ranking can be used to help in pruning redundant rules which is less confidence than general rules. For example if a choice between two and more rules in the rule evaluation step occurs, the rule with the highest rank is selected. This means, specific rules that have lower order than general rules will never be chosen and thus the removed since more general rules have covered all objects matching their body in the evaluation step. Generally, rule ranking in AC is based on support, confidence and cardinality of the rule's antecedent.

3.4.3 Rule Pruning

One of the significant steps in AC mining is cutting down unnecessary rules that may lead to incorrect prediction [133]. This step usually happens once all rules are discovered and sorted where a procedure or more are called to prune redundant rules. For each rule that is sorted, the algorithm evaluates the applicability after beginning with the first rule, which is against the training case. However, if it partially matches at least one training case that will get the rule inserted into the classifier. Actually, most of the current AC algorithms like MMAC [46] and CBA[43] insert a rule into the classifier rule even the rule pruning minimize over fitting; that will happen if it

has the same class as the training case and it matches the training case. The matching between class labels of the candidate rule and the training case does not necessarily give an additional indication of rule goodness besides the matching condition between the rule bodies. We argue that matching between the candidate rule and the training case even if that matching is partial may not totally affect the predictive power of the resulting classification models during the prediction step. In section 4.5 describe the entire proposed rule pruning.

3.4.4 Predicting of Test Data

In this section, we discuss the proposed prediction method which takes into consideration two main thresholds associated with a rule (rule confidence and rule support) as a means to distinguish a group of rules that are applicable to the test data case.

To produce the classifiers, the researcher has used cross validation to logically split the data. The cross validation method divides the training data set into $(n+1)$ folds arbitrary and the rules get learned from n folds in each iteration and then evaluated on the remaining hold out fold. The process is repeated $n+1$ times and the results are averaged and produced. In the experiments, we have set the number of folds in cross validation to 10 similar to other research studies [43] [133].

Furthermore, in the data mining, the basic objective for classification task is predicting the class labels of a previous unseen data (test data). Moreover, it is divided into two groups, first rule, and prediction procedure. The first rule is

applicable to the test case classifiers. The prediction procedure is based on one rule like those used in MCAR and CBA. Thus, rules prediction group which contain algorithms including CPAR [35] and CMAR [102], as well as, the test cases is predicting after they scored based on methods that are used for group of rules. Actually, there is more than one rule contributing to the last decision by using group of rules for prediction. This presents a better chance to condition the test cases satisfying through about a single rule to predict. Hence, the prediction step may at times produce good classifiers by utilizing algorithms that use one rule.

The prediction algorithm which developed through this study use group rule prediction that divided into groups for each class; and then calculate the average support values as well as the averages confidence values for each group in the proposed rule prediction and predicting test data case. Lastly, it assigns the largest average confidence with test case the class of the group. In general, the prediction method considers as the largest average support group, where the cases are groups with similar average with two or more groups. This method ensures a large number of rules during the matching process and therefore the class assignment decision is based on multiple rules rather than single rule as in CBA and MCAR algorithm. In section 4.6 describe the entire proposed rule prediction.

3.5 Development Classifier and Evaluation

The proposed AC algorithm is implemented using (Visual Basic VB) and appendix D includes some screen shot of the developed system development. Comparison study is based on our implementation as well as experiments undertaken using data mining package WEKA [137] and CBA [43]. The reason for utilizing WEKA and

CBA is due to the fact that other AC and traditional algorithms are already implemented in these packages so we do not have to re-invent the wheel and implement them again. Chosen algorithms for comparison based on the most prevalent in the field of text mining and achieve the highest published results [43,106,111].

Experiments on the different data sets from the UCI [76] and Reuters -21578 [69] data collection is conducted using five metrics including prediction accuracy, number of rules, Win\Loss\Tie record, the variation between AC algorithms and CPU computational time.

3.6 Summary

In this chapter, the methodology which is selected for this study is presented. The methodology consists of five phases; the data collection is the first phase of the methodology where in this phase the unstructured data is gathered, the data collection procedures and gathered data for testing our proposed model were reported. The second phase which is the design of the classifier model and it contains rule discover after that comes the third phase which develop the model to get a classifier model in the end of it, the fourth phase was to produce new test data and the last phase is to test the accuracy. Details of the experiments which were conducted are also given, in order to understand the way of evolution. The steps of developing a new model with more precise predictions were incorporated, in order to develop models that enhance the association classification mining. Rule discovery, Rule ranking, Rule pruning and Rule prediction methods were performed.

CHAPTER FOUR

MODIFIED MULTICLASS ASSOCIATION RULE CLASSIFIER

4.1 Introduction

In this chapter, an algorithm named “Modified Multiclass Association Rule” (mMCAR) is proposed which reduces the number of rules produced by the classifier. mMCAR employs a new class assignment method which resulted only relevant rules are used to predict test cases. On other hand, the rule pruning method considers different scenarios when evaluating rules on the training data set during the process of constructing the classification system. Details of the proposed classifier are presented in the following sub sections.

4.2 Proposed Classifier

The mMCAR goes through three main phases: training, construction of classifier, and forecasting of new cases as shown in Figure 4.1. During the first phase, it scans the input data set to find frequent items in the form $\langle \text{AttributeValue}, \text{class} \rangle$ of size 1. These items are called one-items. Then the algorithm repeatedly joins them to produce frequent two- items, and so forth. It should be noted that any item that appear in the input data set less than the MinSupp threshold gets discarded.

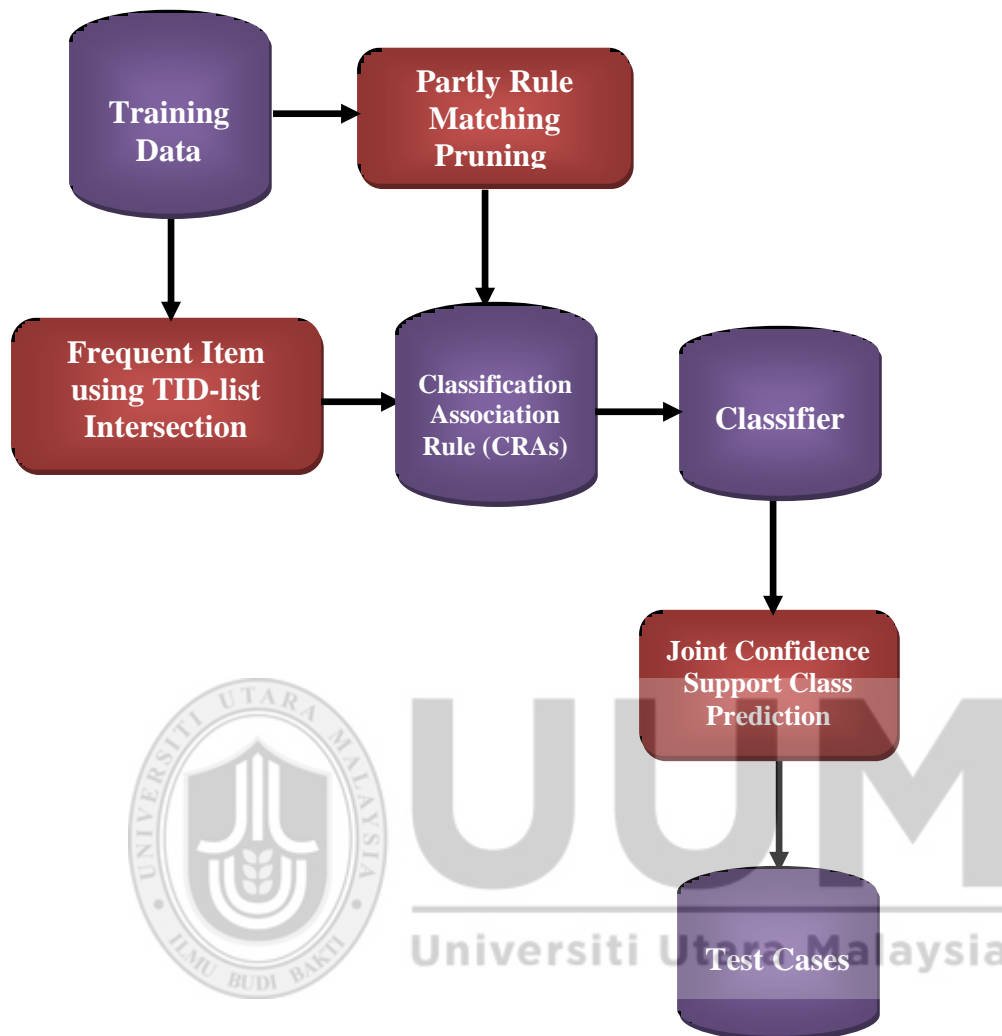


Figure 4.1: mMCAR Steps

The algorithm of mMCAR would check the confidence values all frequent items of all sizes after discovered in first step. CAR will hold a confidence value larger more than the MinConf threshold. On other hand, the CARs represents items that hold high confidence values and statistically representative - if the item gets deleted, and when they completed the set in the training data set. The next step is to sort the rules according to certain

measures and choose a subset of the complete set of CARs to form the classifier.

After the rule is sorted, it gets inserted into the classifier if it covers at least one case full or part match to that of the training case, when the similarity of class unnecessary. In the last step, the algorithm divides all rules into two groups to predict the test case. The mMCAR algorithm is presented in Figures 4.2.

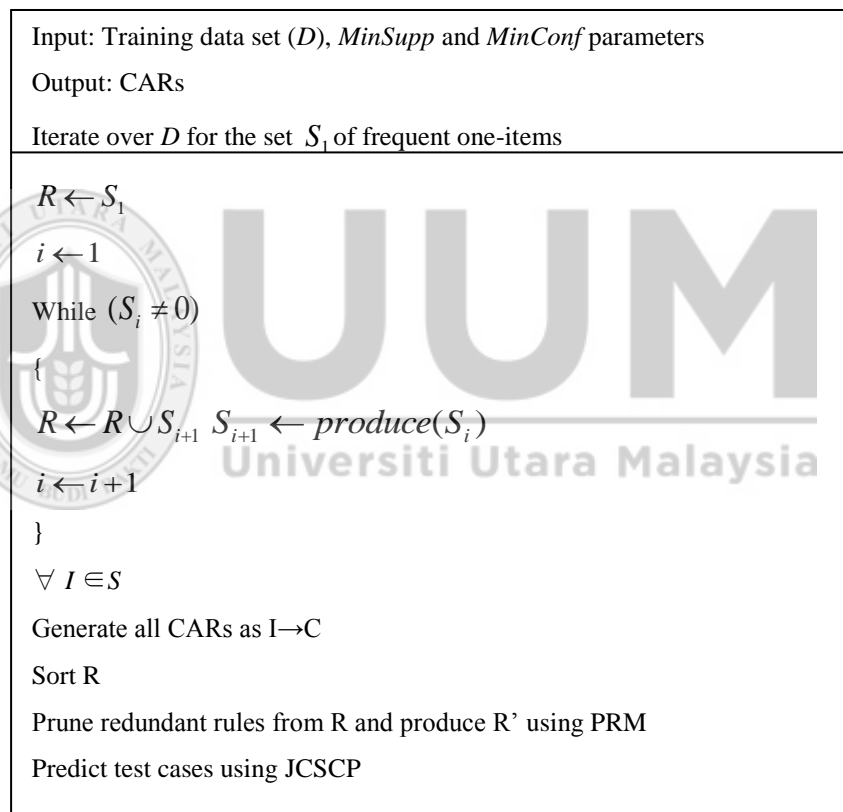


Figure 4.2: The mMCAR Algorithm

4.3 CARs Discovery and Production

The mMCAR uses an intersection method based on what is called Tid-list to compute the support and confidence values of item values. The Tid-list of an item representing the number of rows in the training data set in which an item has occurred. Thus, by intersecting the Tid-lists of two disjoint items,

the resulting set denotes the number of rows in which the new resulting item has appeared in the training data set, and the cardinality of the resulting set represents the new item support value. Such method of computing support of all items without scanning (going through) the training data set for several times is represented as in Figure 4.3.

The vertical mining using vertical and horizontal layout is a training approach and has been used successfully in association rule discovery, i.e. [144], and few years ago in classification, i.e. [37, 145]. This approach transforms the training data set into items Table that contains the locations (Tid-lists) of each item in the training data set, and then it employs simple intersections among these locations to discover frequent values and produce the rules. Since this approach iterates over the training data set only one time therefore it is highly efficient according to experimental studies in the literature with regards to processing time and memory utilization. Once all items of all sizes are discovered, then mMCAR checks their confidence values in a straightforward manner and generate those which pass the MinConf threshold as CARs.

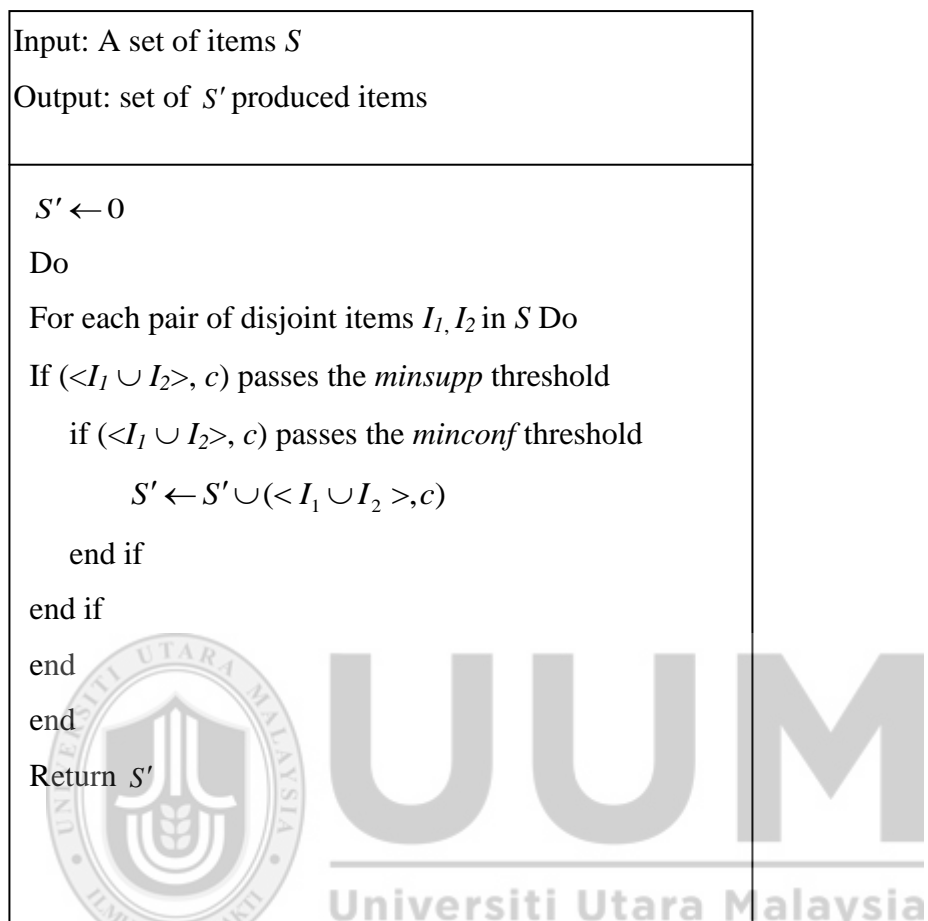


Figure 4.3: Production of Rule

The mMCAR algorithm goes over the training data set only once to count the frequencies of one-items, from which it discovers those that passes the MinSupp. During the scan, frequent one-items are determined, and their appearances in the input data (Tid-lists) are stored inside a data structure in a vertical format. Also, any items that did not pass the MinSupp are removed. Then, the Tid-lists of the frequent one-item are used to produce the candidate two-item by simply intersecting the Tid-lists of any two disjoint one-items.

Consider for instance, the frequent attribute values (size 1) ($\langle a_1 \rangle, I_1$) and ($\langle a_2 \rangle, I_1$) that are shown in Table 4.2 can be utilized to produce the frequent item (size 2) ($\langle a_1, a_2 \rangle, I_1$) by intersecting their Tid-lists, i.e. (1,3,7,8,10) and (1,6,8,10) within the training data set in Table 4.1. The result of the above intersection is the set (1,8,10) which its cardinality equals 3, denoting the support value of the new attribute value ($\langle a_1, a_2 \rangle, I_1$). Now, since this attribute value support is larger than or equal the "MinSupp threshold, 15%, this 2-item will become frequent.

Table 4.1
Training Data Set

RowNo	Attribute ₁	Attribute ₂	class
1	a ₁	a ₂	l ₁
2	a ₁	a ₂	l ₂
3	a ₁	b ₂	l ₁
4	a ₁	b ₂	l ₂
5	b ₁	b ₂	l ₂
6	b ₁	a ₂	l ₁
7	a ₁	b ₂	l ₁
8	a ₁	a ₂	l ₁
9	c ₁	c ₂	l ₂
10	a ₁	a ₂	l ₁

Table 4.2
Frequent Items

Frequent Items			
Rule Condition	Rule class	Supp	Conf
<a ₂ >	I ₁	4/10	4/5
<a ₁ >	I ₁	5/10	5/7

Table 4.3 shows an example data from weather dataset. The parameters of MinSupp and MinConf threshold were set to 15% and 50% respectively. If the MinSupp and MinConf in the example equal or more than the MinSupp and MinConf threshold will pass (not deleted) as a rule.

Table 4.3
Example Data from Weather Dataset

	Outlook	Temperature	Humidity	Play/Class
1	sunny	hot	high	no
2	sunny	hot	high	no
3	overcast	hot	high	yes
4	rainy	mild	high	yes
5	rainy	cool	normal	yes
6	rainy	cool	normal	no
7	overcast	cool	normal	yes
8	sunny	mild	high	no
9	sunny	cool	normal	yes
10	rainy	mild	normal	yes
11	sunny	mild	normal	yes
12	overcast	mild	high	yes
13	overcast	hot	normal	yes
14	rainy	mild	normal	no

Table 4.4 shows the MinSupp and MinConf for one ruleitem class “YES”.

The highlighted ruleitem will be deleted in Table 4.4. The algorithm will

keep the frequent rule item.

Table 4.4
Candidate 1-RuleitemYES

	Attribute	Support	Confident
1	sunny	0.14	0.4
2	overcast	0.28	1
3	rainy	0.21	0.6
4	hot	0.14	0.5
5	mild	0.21	0.5
6	cool	0.21	0.75
7	High	0.21	0.5
8	normal	0.42	0.75

Table 4.5 tabulates the MinSupp and MinConf for one ruleitem class “NO”.

The highlighted ruleitem will be deleted in Table 4.4. The algorithm will

keep the frequent rule item.

Table 4.5
Candidate 1-Ruleitem NO

	Attribute	Support	Confident
1	sunny	0.21	0.6
2	overcast	0	0
3	rainy	0.14	0.4
4	hot	0.14	0.5
5	mild	0.14	0.5
6	cool	0.07	0.25
7	High	0.21	0.5
8	normal	0.14	0.25

Table 4.6 tabulates the MinSupp and MinConf for two ruleitem class “YES”. The highlighted ruleitem will be deleted in Table 4.5. The algorithm will keep the frequent rule item.

Table 4.6
Candidate 2-Ruleitemclass YES

	Attribute	Support	Confident
1	overcast ^ rani	0	0
2	overcast ^ mild	0.07	1
3	overcast ^ cool	0.21	1
4	overcast ^ High	0.14	1
5	overcast ^ normal	0.14	1
6	rainy ^ mild	0.14	0.66
7	rainy ^ cool	0.07	0.5
8	rainy ^ High	0.07	1
9	rainy ^ normal	0.14	0.5
10	mild ^ High	0.14	0.66
11	mild ^ normal	0.14	0.66
12	cool ^ High	0	0
13	cool ^ normal	0.21	0.75

Table 4.7 counts the MinSupp and MinConf for two ruleitem class “NO”. All ruleitem highlighted will be deleted in Table. The algorithm will keep the frequent rule item (the one not highlighted).

Table 4.7
Candidate 2-Ruleitemclass No

	Attribute	Support	Confident
1	Sunny^high	0.21	0.5

Table 4.8 rule generations, will keep the pass frequent rule item from one and two item in all classes.

*Table 4.8
Frequent Items*

RuleID	RuleDesc	Rule Support	Rule Confidence
1	overcast → YES	0.28	1
2	rainy → YES	0.21	0.6
3	mild → YES	0.21	0.5
4	cool → YES	0.21	0.75
5	normal → YES	0.42	0.75
6	sunny → NO	0.21	0.6
7	High → NO	0.21	0.5
8	overcast ^ cool → YES	0.21	1
9	cool ^ normal → YES	0.21	0.75
10	Sunny ^ high → NO	0.21	0.5

4.4 Rule Ranking

In order to give a higher quality rule, the rules must be sorted. This will allow rules with higher priority to be chosen as part of the classifier.

Through this study, the rules sorted according to the following point [41]:

- 1) The rule with higher confidence is placed in a higher rank.
- 2) If the confidence values of two or more rules are the same, then the rule with higher support gets a higher rank.
- 3) If the confidence and the support values of two or more rules are the same, the rule with less number of attribute values in the antecedent gets a higher rank.
- 4) If all above criteria are similar for two or more rules then the rule which was produced first gets a higher rank”.

For each sorted rule (CAR), mMCAR applies it on the training data set. Table 4.9 shows the data from weather dataset after the rule ranking of the guidelines mentioned is applied.

Table 4.9
Rule Ranking

RuleID	RuleDesc	Rule Support	Rule Confidence	Rule Rank
1	overcast → YES	0.28	1	1
2	rainy → YES	0.21	0.6	6
3	mild → YES	0.21	0.5	8
4	cool → YES	0.21	0.75	4
5	normal → YES	0.42	0.75	3
6	sunny → NO	0.21	0.6	7
7	High → NO	0.21	0.5	9
8	overcast ^ cool → YES	0.21	1	2
9	cool ^ normal → YES	0.21	0.75	5
10	Sunny ^ high → NO	0.21	0.5	10

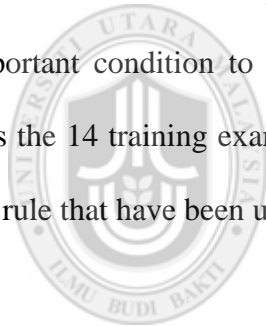
4.5 Pruning Method Partly Rule Match

A rule pruning method proposed in this study for an AC is discussed in this Section. We assume that all candidate rules are extracted and sorted from highest to lowest using confidence, support and rule length criteria.

For each training data, PRM finds the first rule that satisfies the training example by having all of the rule's items inside the training example. When the rule is found, the algorithm marks it and deletes the training example. However, when there is not any

rule that fully matches the training example (do not have a body that could be inside the training).

Whereas the PRM ignores the class similarity as it aims to reduce overlearning. Table 4.10 shows PRM takes on the first rule that partly covers the training example rather than leaving this example to be covered later by the default class rule. By doing this, PRM rule pruning method minimizes the number of training examples that will be used to make the default rule. The main difference between the PRM and of database is that the proposed PRM method includes not only full covered rules but also the partly covered rules into the model. In addition, existing pruning method considers class similarity between the training example and the candidate rule as an important condition to cover the training example and candidate rules. Table 4.9 lists the 14 training examples. Please note that the last column of Table 4.9 denotes the rule that have been used by our method (Classifier).



UUM
Universiti Utara Malaysia

Table 4.10
Rule Pruning Using Weather Dataset

	Outlook	Temperature	Humidity	Play/Class	Classifier
1	sunny	hot	high	no	7
2	sunny	hot	high	no	7
3	overcast	hot	high	yes	1
4	rainy	mild	high	yes	2
5	rainy	cool	normal	yes	6
6	rainy	cool	normal	no	5
7	overcast	cool	normal	yes	1
8	sunny	mild	high	no	7
9	sunny	cool	normal	yes	6
10	rainy	mild	normal	yes	6
11	sunny	mild	normal	yes	6
12	overcast	mild	high	yes	1
13	overcast	hot	normal	yes	1
14	rainy	mild	normal	no	6

Table 4.11
Frequent Item and Rule Ranking for Weather Dataset

RuleID	RuleDesc	Rule Support	Rule Confidence	Rule Rank
1	Overcast → YES	0.28	1	1
2	Rainy → YES	0.21	0.6	6
3	Mild → YES	0.21	0.5	8
4	Cool → YES	0.21	0.75	4
5	Normal → YES	0.42	0.75	3
6	Sunny → NO	0.21	0.6	7
7	High → NO	0.21	0.5	9
8	overcast ^ cool → YES	0.21	1	2
9	cool ^ normal → YES	0.21	0.75	5
10	Sunny ^ high → NO	0.21	0.5	10

Applying the partial rule match pruning method on Table 4.10 and Table 4.11, the training data on four cases (#1,2) have a full match with rule ID (R7) which the rule used to cover the data and delete. The same thing occurs for training data (#3,7,12,13) in which Rule ID 1 used to cover the data and delete. Also training data (#6) have a full Match with rule ID (R5) and is deleted. Training data (#6, 9, 10, 11) in which RuleID 6 has been used to cover and delete the data. Training data (#8) in which RuleID 7 has been used to cover and delete the data. Training data (#14) have full rule match with no similar class in another methods will use the default class in the classifier but using PRM pruning method use Rule ID 6 to cover and delete the data. The PRM pruning method terminates when all candidate rules are tested and the training dataset is empty.

The above example shows the demonstration of the proposed rule pruning method that indeed reduces error by allowing partly matching rule to be part of the classifier instead on taking the default rule. All rules that have been applied during the classifier builder are inserted into the classifier whereas the remaining rules get deleted since they have no training data coverage. In summary, the proposed PRM is as shown in Figure 4.4. The input of the PRM method is the training data (TranD) and discovered Rules Rank is (RuleR). And, the output is classifier (C).

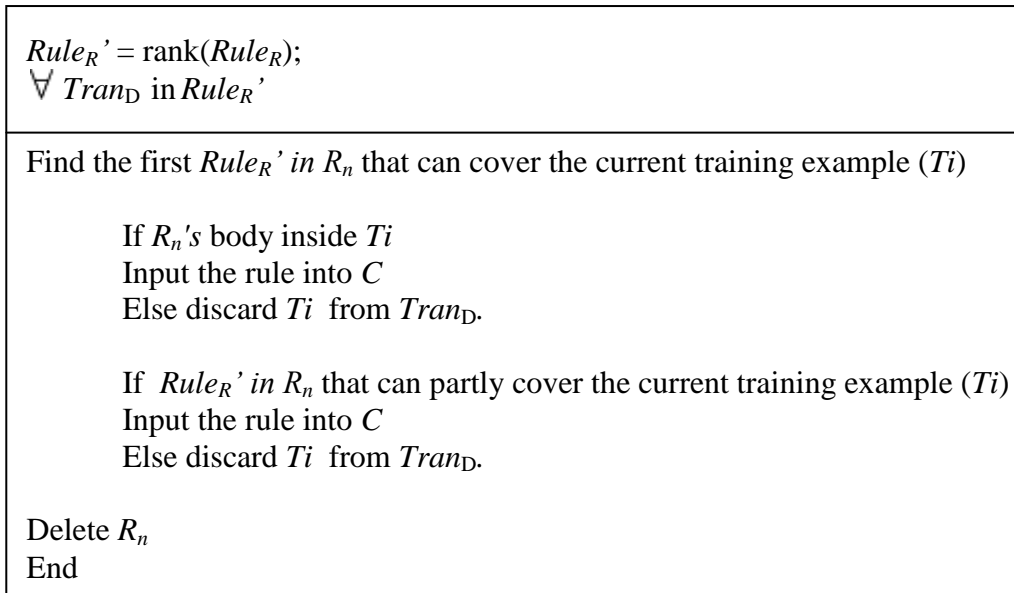


Figure 4.4: Partly Rule Match Pruning Method

4.6 Joint Confidence Support Class Prediction Method

In data mining, the basic objective for classification task is predicting the class labels of an unseen data (test data). Moreover, it is divided into two groups, namely, first rule, and prediction procedure. The first rule is applicable to the test case classifiers. The prediction procedure is based on one rule similar to those used in MCAR and CBA. Thus, rules prediction group is contains algorithms including CPAR [35] and CMAR [10], as well as, the test cases is predicted after they score based methods are used for group of rules. Actually, there is more than one rule contributing to the last decision when using group of rules for prediction. This resulted in a better chance to condition the test cases satisfying through about a single rule to predict. Hence, the prediction step may at times produce good classifiers by utilizing algorithms that use one rule.

JCSCP shown in Figure 4.5 is a class allocation method based on joint probabilities of the rules that are applicable to the test data. This method assumes that the classification model is AC based. When a new test data is about to be forecast, the proposed prediction method iterates over the rules in the model and finds the all rules applicable to the test data. Now if all rules predict the same class then the JCSCP method basically assigns that class to the test data on a straight forward manner. In cases that applicable rules have different class labels, the JCSCP splitd those into groups based on the class value and for each group it computes its weight: The weighted average of multiplying the rules supported with the rules confidence can be computed based on the equation below:

$$GroupWeight = \sum_{i=1}^k \frac{(R_k Support * R_k Confidence)}{|K|} \quad (4.1)$$

Where: R_k = rule weight

Basically, for each group of applicable rules, the JCSCP multiplies the support and confidence of each rule belonging to the group and then sums up all values for all rules in the same group and divides that with the number of rules in the group. The class belongs to the group that have the highest result is then assigned to the test case. In this case, two important factors have been considered by our prediction method:

- 1) All applicable rules have contributed to assigning the class to the test case instead of just a single rule.
- 2) Two important parameters associated with the rule has also played a crucial role in the process.

Table 4.12 shows the set of rules which is derived through the progress mMCAR with 20% in Minsup and 40% in Minconf. Figure 4.5 illustrates the idea of the proposed class prediction method, which is to choose the majority class between set of the representative, general rules, and highest confidence and in the set of rules R to predict Ts in the test data. The proposed method classifies a test case, divides the rules that are applicable to Ts into several groups based on the class label. Then it calculates the average confidence and support for each group. Lastly, the class that belongs to the sum of the largest average support and confidence is given to Ts. In cases where there is no rule that matches the Ts condition, the default class will be assigned to the test Ts.

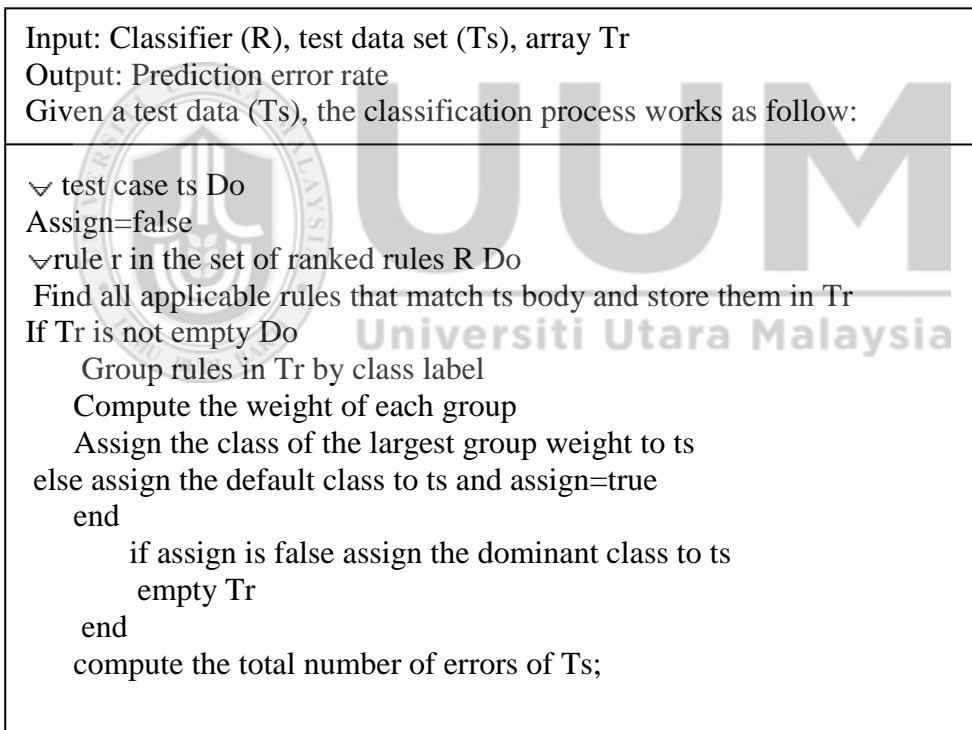


Figure 4.5: *Joint Confidence Support Class Prediction*

Table 4.12
A Rule-Based Model

RuleID	RuleDesc	Rule Support	Rule Confidence	Rule Rank
1	overcast → YES	0.28	1	1
2	rainy → YES	0.21	0.6	6
3	mild → YES	0.21	0.5	8
4	cool → YES	0.21	0.75	4
5	normal → YES	0.42	0.75	3
6	sunny → NO	0.21	0.6	7
7	High → NO	0.21	0.5	10
8	overcast ^ cool → YES	0.21	1	2
9	cool ^ normal → YES	0.21	0.75	5
10	Sunny ^ high → NO	0.21	0.5	11

Table 4.13
Testing Case

	Outlook	Temperature	Humidity	Actual Class	Predicted Class
1	sunny	mild	normal	yes	Yes
2	sunny	hot	high	no	ON
3	rainy	mild	high	yes	Yes
4	sunny	hot	high	no	Yes
5	rainy	cool	normal	yes	Yes
6	overcast	mild	high	yes	Yes

The method can be described through Table 4.13 and Table 4.14. The test case is shown in Table 4.13, which are applicable to ts". Now, to classify ts, we count the Applicable rules per class, we found that YES class is the largest count so we predict class YES" for Ts.

Table 4.14

Applicable Rules for Ts

	Outlook	Temperature	Humidity	Actual Class	support	confidant	Average
1	sunny	mild	normal	yes	0.64	1.25	0.40
2	sunny	hot	high	no	0.42	1.10	0.23
3	rainy	mild	high	yes	0.42	1.10	0.23
4	Sunny	hot	high	no	0.42	1.10	0.23
5	rainy	cool	normal	yes	0.84	2.13	0.89
6	overcast	mild	high	yes	0.49	1.5	0.74

4.7 Summary

In this chapter, a new classification based association rule algorithm called mMCCAR has been proposed. This algorithm employs a new classifier building method that limits the use of redundant and misleading rules from taking any part in the prediction step. This chapter introduced three new methods to enhance the mMCCAR accuracy and number of rules. New rule discovery is performed to reduce the number of rule generation, new rule pruning with partly rule match and ignores the class similarity to more accurate result and reduce overlearning. Finally new rule prediction to assigning the class to the test case instead of just a single rule.

CHAPTER FIVE

RESULTS AND DISCUSSION

5.1 Introduction

This chapter presents the results obtained via experiments conducted on structured and unstructured data. The experiment on structured data is performed on UCI data sets and uses different classification learning algorithms (C4.5, RIPPER, CBA, and MCAR) in order to evaluate the effectiveness of mMCAR.

The experiments on unstructured data is performed on Reuters-21578 data sets using different classification learning algorithms (CBA, BCAR and MCAR from association classification (AC) Naïve Bayes, K-NN and SVM) to evaluate the effectiveness of mMCAR.

The main parameters of mMCAR, MinSupp and MinConf were set to 2% and 50% respectively in the experiments. Hence, the support threshold is the key factor which controls the number of rules produced in AC. Based on that, the number of extracted rules will be small if the support value is high. All experiments were executed on Pentium IV machine with 2.0 GB RAM and 2.6 GH processor. We have implemented mMCAR using VB.net and MCAR using Java, and the results of RIPPER and C4.5 were derived from WEKA [137], an open source machine learning tool.

5.2 Rules Obtained Using Both the MCAR and mMCAR

In this section, the behaviour of the new mMCAR algorithm is explained in comparison with MCAR algorithm, which contains three attributes as well as the class attribute. It is essential to understand how to minimize the number of rules when it covers all data between both algorithms manually. In Table 5.1, the difference between the pruning methods of mMCAR algorithm compared to MCAR algorithm is shown. For the sake of argument let us assume that the MinSupp and the MinConf have been set to 20% and 40% respectively for presentation purposes.

*Table 5.1
Training Data Set*

Attribue1	Attribue2	Attribue3	Class	Rule applied by MCAR	Rule applied by mMCAR
x ₁	y ₁	z ₁	class ₂	R1	R1
x ₁	y ₁	z ₂	class ₂	R1	R1
x ₂	y ₁	z ₂	class ₁	Default	R1
x ₁	y ₂	z ₂	class ₁	R1	R1
x ₃	y ₁	z ₁	class ₁	Default	R1
x ₁	y ₁	z ₁	class ₂	R1	R1
x ₂	y ₂	z ₃	class ₁	R2	R2
x ₁	y ₂	z ₃	class ₁	R2	R1
x ₁	y ₂	z ₂	class ₁	R8	R1
x ₁	y ₂	z ₂	class ₂	Default	R1

We have applied both mMCAR and MCAR on the training data set depicted in Table 5.1. The rule discovery phase for both algorithms terminates in the step of 3-rule items. Once that has occurred, both algorithms compute the confidence for all sets of frequent rule items found to generate those which have enough confidence, e.g. pass MinSupp threshold, as candidate rules. Candidate rules derived by the MCAR and our algorithm are shown in Table 5.2. All other rules are removed by the algorithm and thus MCAR has discovered only eight candidate rules from Table 5.2.

Table 5.2
 Ranked Candidate Rules Produced by MCAR and mMCAR

Rule rank	Candidate rule	Class Label	Support Frequency	Confidence	MCAR classifier	mMCAR classifier
1	$(x_1, y_1) \rightarrow \text{class}_2$	c_2	3	100.00 %	$(x_1, y_1) \rightarrow \text{class}_2$	$(x_1, y_1) \rightarrow \text{class}_2$
2	$z_3 \rightarrow \text{class}_1$	c_1	2	100.00 %	$z_3 \rightarrow \text{class}_1$	$z_3 \rightarrow \text{class}_1$
3	$x_2 \rightarrow \text{class}_1$	c_1	2	100.00%	$x_2 \rightarrow \text{class}_1$	$x_2 \rightarrow \text{class}_1$
4	$(x_1, z_1) \rightarrow \text{class}_2$	c_2	2	100.00 %	$(x_1, z_1) \rightarrow \text{class}_2$	$(x_1, z_1) \rightarrow \text{class}_2$
5	$(y_2, z_3) \rightarrow \text{class}_1$	c_1	2	100.00 %	$(y_2, z_3) \rightarrow \text{class}_1$	$(y_2, z_3) \rightarrow \text{class}_1$
6	$(x_1, y_1, z_1) \rightarrow \text{class}_2$	c_2	2	100.00 %	$(x_1, y_1, z_1) \rightarrow \text{class}_2$	$(x_1, y_1, z_1) \rightarrow \text{class}_2$
7	$y_2 \rightarrow \text{class}_1$	c_1	4	80.00%	$y_2 \rightarrow \text{class}_1$	$y_2 \rightarrow \text{class}_1$
8	$(x_1, y_2) \rightarrow \text{class}_1$	c_1	3	75.00%	$(x_1, y_2) \rightarrow \text{class}_1$	$(x_1, y_2) \rightarrow \text{class}_1$
	Default	c_1			Default	Default

As soon as the candidate rules are generated, they get ranked according to the ranking parameter described in Section 4.4, e.g. Confidence, support, and rule's number of attribute values (the less the better). The rules after ranking are depicted in Table 5.2. In rule pruning, and for each training data excluding the last column, our algorithm iterates over the rules (top down) and selects the rule that is partly contained inside the training data. So, for our algorithm, rule #1 covers most training cases of Table 5.1, and rule #2 covers 1 training data. We end up having 2 rules classified from Table 5.1. For MCAR, this algorithm is conservative and requires full class similarity between the rule body and the attribute(s) in the training data set in addition to identical class. Three rules have covered the training data for MCAR and the algorithm was forced to generate a default rule for all data that was unclassified. Both MCAR and our algorithm's rules are shown in the last 2 columns of Table 5.2. This example proves that our pruning method:

- 1) covers more training data per rule and therefore produces smaller classifiers than MCAR
- 2) Reduces the utilisation of the default rule which usually may cause high errors during prediction

Table 5.3
Sample of Rules by mMCAR and MCAR on UCI “Cleve” Data Set

<i>mMCAR</i>	<i>MCAR</i>
$X \rightarrow Cleve$	$X \rightarrow Cleve$
$Y \rightarrow Cleve$	$Y \rightarrow Cleve$
$fal \rightarrow Cleve$	$fal \rightarrow Cleve$
$naotang \rightarrow Cleve$	$naotang \rightarrow Cleve$
$X \& naotang \rightarrow Cleve$	$X \& naotang \rightarrow Cleve$
$Y \& naotang \rightarrow Cleve$	$Y \& naotang \rightarrow Cleve$
$X \& fal \rightarrow Cleve$	$X \& fal \rightarrow Cleve$
$Y \& naotang \rightarrow Cleve$	$Y \& naotang \rightarrow Cleve$
	$naotang \& fal \rightarrow Cleve$

Table 5.3 contains sample of rules for the UCI “Cleve” data set produced using mMCAR and MCAR. The Table shows that mMCAR produced one rule less than the MCAR, that is mMCAR uses only 8 rules while MCAR employs 9 rules. The $naotang \& fal \rightarrow Cleve$ is not included in mMCAR because by using the Partly Rule Matched pruning method, the match between the data and the rules was handled by the 8 earlier rules.

Table 5.4 presents sample of rules obtained by mMCAR and MCAR for Reuter’s “acq” data set. The Table includes 8 rules and 10 rules for mMCAR and MCAR respectively. The $Company \& year \rightarrow acq$ and $Year \& stake \rightarrow acq$ are the two rules that were not included in mMCAR. This is due to the new rule pruning method that has found the match for all the data using limited size of rules. The next two sections will present the results of the structured (UCI) and unstructured Reuter’s data set.

Table 5.4
Sample of rules by mMCAR and MCAR on Reuter's "acq" data set

mMCAR	MCAR
shares→acq	shares→acq
company→acq	company→acq
year→acq	year→acq
stake→acq	stake→acq
Shares& company →acq	Shares& company →acq
Shares& year →acq	Shares& year →acq
Shares& stake →acq	Shares& stake →acq
Company& stake →acq	Company& year →acq
	Company& stake →acq
	Year& stake →acq

5.3 Structured Data Set

In this section, results from different traditional classification algorithms as well as rule-based classification algorithms are compared with mMCAR based on the prediction accuracy, number of rules, win-tie-loos record, and Compression variation between AC algorithms and CPU time. For the experiments, fifteen UCI data sets are used [76], and the algorithms tested for the comparison are the MCAR [133], C4.5 [41], RIPPER [146] and CBA [43]. The reason behind selecting these algorithms is the different training strategy used in discovering the rules. For example, C4.5 employs divide and conquer while RIPPER utilises heuristic based strategy. On the other hand, MCAR employs associative classification.

5.3.1 Prediction Accuracy for UCI Data Set

The prediction accuracy of the proposed algorithm as well as RIPPER, C4.5, CBA, and MCAR is shown in Table 5.5. Data in the Table shows that mMCAR, MCAR and C4.5 achieve consistent accuracy. The comparison between mMCAR and RIPPER has shown that the results in mMCAR were

more accurate in eleven data sets compared with RIPPER which shown more accuracy in (*Breast, Iris, Mushroom and Vote*) of the data sets. The comparison between mMCAR and C4.5 has shown that the results in mMCAR were more accurate in eight data sets compared with C4.5 which shown more accuracy in (*Breast, Glass, Labor, Led7, Lymph and Pima*) of the data sets. The comparison between mMCAR and CBA has shown that the results in mMCAR were more accurate in nine data sets compared with CBA which shown more accuracy in (*Breast, Cleve, Labor and Win*) of the data sets. The comparison between mMCAR and MCAR has shown that the results in mMCAR were more accurate in six data sets compared with MCAR which shown more accuracy in (*Balance-Scale, Breast, Cleve, Heart-s, Mushroom and Pima*) of the data sets. The mMCAR algorithm achieved good accuracy on overall data sets. The mMCAR got more accurate results than algorithms (RIPPER, C4.5 and CBA) of the most data sets; on the other hand the same results in the experiment were drawn from (MCAR and mMCAR).

Table 5.5
The Prediction Classification Accuracy on UCI Data Sets

Data set	RIPPER	C4.5	CBA	MCAR	mMCAR
Austra	85.2%	86.4%	85.4%	86.1%	86.4%
Balance-scale	74.6%	64.9%	68.2%	77.0%	76.2%
Breast	95.4%	93.6%	94.7%	95.0%	93.8%
Cleve	77.6%	77.3%	83.1%	81.8%	78.9%
Glass	68.7%	77.6%	69.9%	71.4%	74.2%
Heart-s	78.2%	79.1%	71.2%	81.2%	80.5%
Iris	94.7%	94.7%	93.3%	92.9%	94.7%
Labor	77.2%	85.0%	95.0%	83.5%	83.5%
Led7	69.5%	73.4%	72.4%	71.8%	73.1%
Lymph	77.0%	82.0%	74.4%	78.1%	78.1%
Mushroom	99.9%	99.6%	98.9%	99.6%	99.7%
Pima	73.3%	77.7%	75.5%	77.1%	76.4%
Vote	88.3%	87.8%	87.4%	88.2%	87.4%
Wine	94.4%	93.2%	98.3%	95.7%	95.7%
Weather	64.3%	71.4%	85.0%	84.1%	85.0%

The prediction accuracy of all considered algorithms generated from the data sets under consideration is illustrated in Figure 5.1. It is obvious from this Figure that the rule induction classification approach (RIPPER) has achieved the least accuracy and AC approach (MCAR, mMCAR) achieved the largest prediction accuracy. The main reason for AC achieving high accuracy is attributed to the fact that such approach often investigates the complete correlations between the attribute values and the class attribute. This usually results in numerous a high volume of knowledge not found by traditional classification data mining algorithms.

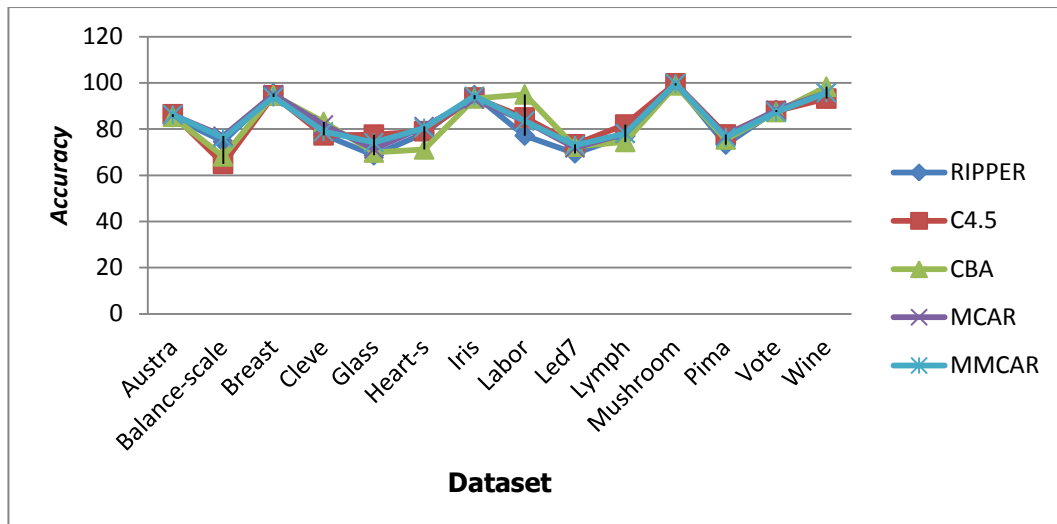


Figure 5.1: Prediction Accuracy on UCI Data sets

mMCAR only considers similarity between the rule body (i.e precedent of a rule or Left Hand Side) and the training data which ensures high data coverage per rule and therefore less number of rules in the classifier. In other words, we try to balance between the size of the classifiers and classification accuracy by allowing a slight loss of accuracy in order to have a smaller set of rules.

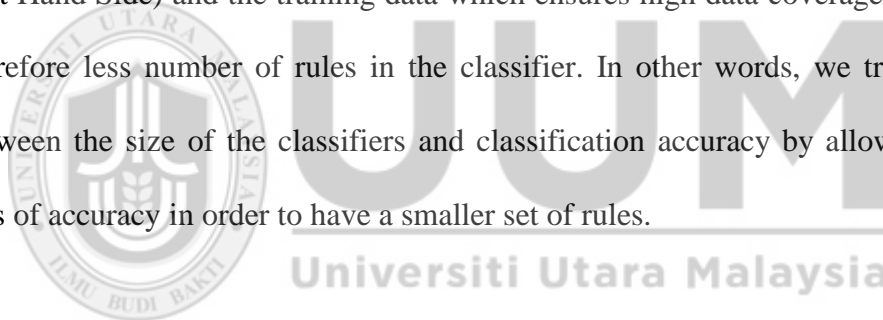


Table 5.6
The Classification Accuracy Between mMCAR and all Algorithms

Data set	RIPPER	mMCAR	C4.5	mMCAR	CBA	mMCAR	MCAR	mMCAR
Austra	85.2%	86.4%	86.4%	86.4%	85.4%	86.4%	86.1%	86.4%
Balance-scale	74.6%	76.2%	64.9%	76.2%	68.2%	76.2%	77.0%	76.2%
Breast	95.4%	93.8%	94.6%	93.8%	94.7%	93.8%	95.0%	93.8%
Cleve	77.6%	78.9%	77.3%	78.9%	83.1%	78.9%	81.8%	78.9%
Glass	68.7%	74.2%	77.6%	74.2%	69.9%	74.2%	71.4%	74.2%
Heart-s	78.2%	80.5%	79.1%	80.5%	71.2%	80.5%	81.2%	80.5%
Iris	94.7%	94.7%	93.7%	94.7%	93.3%	94.7%	92.9%	94.7%
Labor	77.2%	83.5%	85.0%	83.5%	95.0%	83.5%	83.5%	83.5%
Led7	69.5%	73.1%	73.4%	73.1%	72.4%	73.1%	71.8%	73.1%
Lymph	77.0%	78.1%	82.0%	78.1%	74.4%	78.1%	78.1%	78.1%
Mushroom	99.9%	99.7%	99.6%	99.7%	98.9%	99.7%	99.6%	99.7%
Pima	73.3%	76.4%	77.7%	76.4%	75.5%	76.4%	77.1%	76.4%
Vote	88.3%	87.4%	87.8%	87.4%	87.4%	87.4%	88.2%	87.4%
Wine	94.4%	95.7%	93.2%	95.7%	98.3%	95.7%	95.7%	95.7%
Weather	64.3%	85.0%	71.4%	85.0%	85.0%	85.0%	84.1%	85.0%

Table 5.6 showed that the accuracy performance of mMCAR comparing with the entire algorithm used in this experiment. According to the results, mMCAR wins with eleven dataset, while RIPPER gets two dataset more accurate. Moreover, mMCAR get eight dataset, when C4.5 six dataset more accurate. On other hand, mMCAR wins with nine dataset, while CBA get four dataset more accurate. Finally, the mMCAR algorithm has six dataset more accurate, while MCAR has also six dataset more accurate.

5.3.2 Number of Rules for UCI Data Set

Number of rules obtains by the proposed algorithm as well as RIPPER, C4.5, CBA, and MCAR is shown in Table 5.7. The comparison between mMCAR and RIPPER has shown that the results in mMCAR were achieved better result in nine data sets compared with RIPPER which achieved better result in (*Balance-scale, Breast, Heart-s* and *Lymph*) of the data sets. In other hand the comparison between mMCAR and C.45 has shown that the results in mMCAR were achieved better result in six data sets compared with C.45 which achieved better result in (*Balance-scale, Breast, Heart-s* and *Lymph*) of the data sets. The comparison between mMCAR and CBA has shown that the results in mMCAR were achieved better result in seven data sets compared with CBA which achieved better result in (*Austral, Cleve, Led7, Lymph, Mushroom, Pima, and Vote*) of the data sets. The comparison between mMCAR and MCAR has shown that the results in mMCAR were achieved better result in nine data sets compared with MCAR which achieved better result in (*Heart-s* and *Lymph*) of the data sets. The result indicates that the proposed algorithm derives less number of rules in most cases than considered algorithms.

Table 5.7

The Number of Rules for the UCI Data Sets

Data set	RIPPER	C4.5	CBA	MCAR	mMCAR
Austra	185	63	121	185	163
Balance-scale	17	17	45	19	19
Breast	59	66	78	61	61
Cleve	101	94	72	100	97
Glass	28	35	36	36	27
Heart-s	33	25	52	35	36
Iris	16	11	18	16	11
Labor	15	15	17	15	15
Led7	161	83	53	162	83
Lymph	51	56	38	47	54
Mushroom	48	47	38	42	42
Pima	75	54	36	88	58
Vote	86	84	40	85	74
Wine	11	11	11	12	11
Weather	5	6	6	6	4

An analysis on the number of rules derived by the classifier has been conducted. Figure 5.2 depicts the classifier size extracted for each UCI data sets using RIPPER, C4.5, CBA, MCAR and mMCAR algorithms. So mMCAR algorithm outperformed RIPPER, C4.5, and MCAR on several data sets. When the same results for mMCAR and CBA in seven dataset.

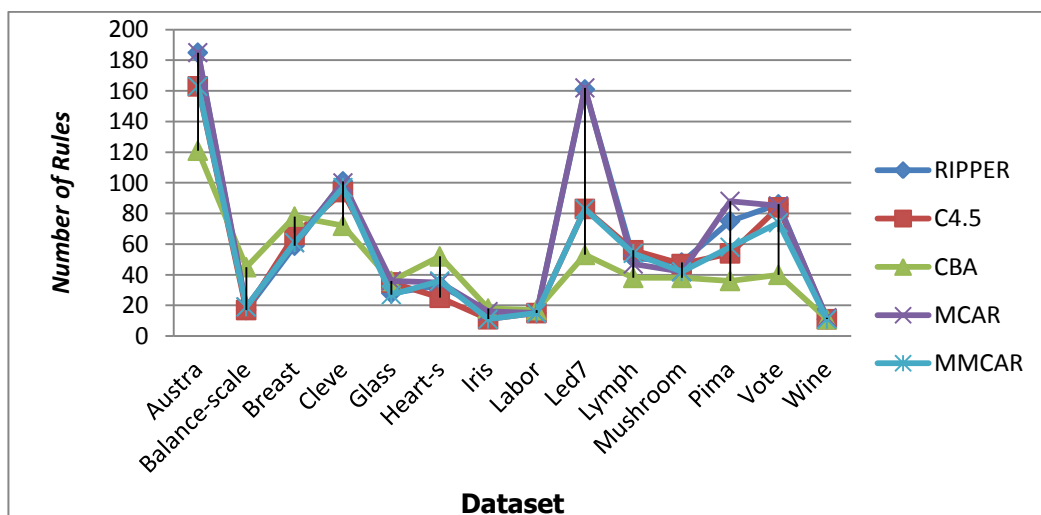


Figure 5.2: Number of Rules of the on UCI Data Sets

The relationship between classification accuracy and number of rules for each UCI data set is a positive linear relation. As shown in Figures 5.1 and 5.2, when the number of rules decreases the accuracy also decreases. This is true for the majority of the data sets.

Table 5.8 The Number of Rules Between mMCAR and all Algorithms

Data set	RIPPER	mMCAR	C4.5	mMCAR	CBA	mMCAR	MCAR	mMCAR
Austra	185	163	63	163	121	163	185	163
Balance-scale	17	19	17	19	45	19	19	19
Breast	59	61	66	61	78	61	61	61
Cleve	101	97	94	97	72	97	100	97
Glass	28	27	35	27	36	27	36	27
Heart-s	33	36	25	36	52	36	35	36
Iris	16	11	11	11	18	11	16	11
Labor	15	15	15	15	17	15	15	15
Led7	161	83	83	83	53	83	162	83
Lymph	51	54	56	54	38	54	47	54
Mushroom	48	42	47	42	38	42	42	42
Pima	75	58	54	58	36	58	88	58
Vote	86	74	84	74	40	74	85	74
Wine	11	11	11	11	11	11	12	11
Weather	5	4	6	4	6	4	6	4

Table 5.8 shows a result for mMCAR comparing to other algorithm used in the experiment. The output confirmed that the mMCAR wins eight dataset, while RIPPER wins in five dataset less number of rules. The mMCAR wins in six dataset, while C4.5 algorithm wins in five dataset. mMCAR has wins in seven dataset when CBA wins also seven dataset. Finally, mMCAR algorithm wins in nine dataset, MCAR wins in only two dataset.

5.3.3 Win-Loss-Tie Record for UCI Data Set

The prime objective of present research study is producing associative classifier that could be highly accurate by using few rules. Table 5.5 depicts the win-loss-tie record for the accuracy metrics of mMCAR algorithm and Table 5.9 demonstrates all considered algorithms. We may deduce that the win-loss-tie record for accuracy of mMCAR against RIPPER, C4.5, CBA and MCAR are 11-4-0, 9-6-0, 8-5-1 and 6-6-3 respectively. The mMCAR algorithm is better than RIPPER and CBA and has outperformed both on numerous data sets in accuracy of prediction and mMCAR is far better than MCAR and C4.5 algorithm based on the number of rules. However, the overall average of accuracy on all the data sets considered for mMCAR is best among other algorithms which demonstrates its consistency in high quality classifiers as per classification accuracy. Alternatively, the win-loss-tie record for number of rules obtained mMCAR against all considered algorithms are depicted in Table 5.10 and we can conclude that the win-loss-tie record of mMCAR against RIPPER, C4.5, CBA and MCAR are 8-5-2, 6-5-4, 7-7-1 and 8-4-2 respectively. Therefore, mMCAR algorithm is proved better than all other algorithms except CBA. In CBA, the seven data sets revealed similar results which is evidence that mMCAR is better than CBA for 7 data sets and opposite also holds true for seven data sets as well.

*Table 5.9
Won Loss-Tie Accuracy for UCI Dataset*

Category/Algorithm	RIPPER	C4.5	CBA	MCAR
mMCAR	11-4-0	8-6-1	9-4-2	6-6-3

*Table 5.10
Won-Loss-Tie Number of Rules for UCI Dataset*

Category/Algorithm	RIPPER	C4.5	CBA	MCAR
mMCAR	9-4-2	6-5-4	7-7-1	9-2-4

The mMCCAR only deliberates the similarity of training data and the rule body as shown in the results that guarantees more and high data coverage per rule, consequently classifier having less number of rules. Stated otherwise, the research focuses to maintain a balance between classification accuracy and the size of the classifiers through permitting a minor loss of accuracy for having a less number of rules classifiers.

5.3.4 Compression Variation Between AC Algorithms

The variations between AC algorithms MCAR, CBA and mMCCAR are represented in Table 5.11 with total variation and variation for all data sets.

*Table 5.11
The Variation of UCI Data Set Between AC Algorithms*

Data set	N.rule mMCCAR vs CBA	Accuracy mMCCAR vs CBA	N.rule mMCCAR vs MCAR	Accuracy mMCCAR vs MCAR	n.rule MCAR vs CBA	Accuracy MCAR vs CBA
Austra	42	1	-22	-0.3	64	1
Balance- scale	-26	8	0	0.8	-26	8
Breast	-17	-0.9	0	1.2	-17	-0.9
Cleve	25	-4.2	-3	2.9	28	-4.2
Glass	-9	4.3	-9	-2.8	0	4.3
Heart-s	-16	9.3	1	0.7	-17	9.3
Iris	-7	1.4	-5	-1.8	-2	1.4
Labor	-2	-11.5	0	0	-2	-11.5
Led7	30	0.7	-79	-1.3	109	0.7
Lymph	16	3.7	7	0	9	3.7
Mushroom	4	0.8	0	-0.1	4	0.8
Pima	22	0.9	-30	0.7	52	0.9
Vote	34	0	-11	0.8	45	0
Wine	0	-2.6	-1	0	1	-2.6
Weather	-2	0	-2	-0.9	0	0
Total	94	10.8	-154	-0.1	248	10.9

Table 5.11 shows the results of the variation between AC algorithm in the experiment. The first column of the Table shows the variation between CBA and the

mMCAR and the results show that mMCAR increase 94 rule. The 4th column of the Table gives the variation between CBA and mMCAR in terms of accuracy and the total result show mMCAR increase the accuracy 10.8. The 3rd column of the Table gives the variation between MCAR and mMCAR in terms of number of rules and the result confirm that mMCAR decrease 154 rules. The 2nd column in the Table demonstrates the variations in the accuracy between mMCAR and MCAR that is almost same with a difference of -0.1 in total. The fifth column in Table above shows the variation between CBA and MCAR in terms of number of rules and this is evident from result that CBA decrease the rule 248 rule. The sixth column in the Table above illustrates the variations between CBA and MCAR in terms of accuracy and the result show that mMCAR increase the accuracy by 10.9.

We can conclude from above discussion and the Table above that the MCAR can increase the total accuracy by 10.9 as compared to CBA while MCAR need to increase 248 rules. On the other hand, the accuracy result of mMCAR is 10.8 but increase 94 rules, while MCAR need 248 rules. This is evidence that the objective is achieved as the number of rules are reduced by maintain the competitive accuracy.

5.3.5 Computational Time for UCI Data Set

This section show the time taken for AC algorithm (RIPPER, C4.5, CBA, MCAR and mMCAR) to building the classifier on 15 dataset on order to compare efficiency. Table 5.12 shows the runtime in second obtained in the experiment. The runtime revealed that mMCAR is faster than RIPPER, C4.5, CBA and MCAR in most data set. The vertical and horizontal intersection method that mMCAR employed to find the rules and avoiding

going over the data multiple time during building the classifier, are responsible for the runtime advantage. For some data sets that have many attributes such as the Mushroom data set, the time required to find the rule items is substantially minimized in our data if compared with CBA and MCAR algorithm. It is obvious from the Table that our method often takes less time to find frequent rules items than MCAR due to the reduction in the number of joins at each iteration and for each data set.

*Table 5.12
Training Time for UCI Data Sets Using AC Algorithm*

Dataset	RIPPER	C4.5	CBA	MCAR	mMCAR
Austra	0.47	0.22	0.28	0.41	0.30
Balance-scale	0.23	0.21	0.41	0.29	0.18
Breast	0.10	0.13	0.16	0.60	0.11
Cleve	0.15	0.13	0.11	0.13	0.12
Glass	0.09	0.07	0.08	0.10	0.06
Heart-s	0.19	0.16	0.21	0.17	0.16
Iris	0.23	0.21	0.27	0.21	0.20
Labor	0.14	0.09	0.14	0.09	0.06
Led7	1.24	0.52	0.37	0.98	0.48
Lymph	0.19	0.15	0.1	0.13	0.12
Mushroom	4.02	3.86	2.67	3.88	1.74
Pima	0.31	0.14	0.08	0.11	0.09
Vote	0.26	0.17	0.09	0.13	0.12
Wine	0.13	0.12	0.12	0.12	0.10
Weather	0.07	0.08	0.07	0.6	0.5

5.4 Unstructured Dataset

The mMCAR is compared with rule-based classification algorithms and different traditional classification algorithms based on the prediction accuracy, number of rules, win-tie-loos record, compression variation between AC algorithms and CPU time. The Reuters-21578[69] is the data used in the experiment. The Reuters-21578 version ModApte comprises 9,174 documents which are divided into 2,571 of testing documents and 6,603 training; An experimenter then develops a categorization system by

automated training on the training set. The algorithms used in the comparison are CBA [43], BCAR [39] and MCAR [37] from the Association classification approaches while Naïve Bayes [77], K-NN [147] and SVM [148] represent the traditional approaches. We tested the proposed algorithm using the minsupp and minconf values of 2%, and 50%, respectively.

5.4.1 Prediction Accuracy for Reuters Data Set

Table 5.16 gives the accuracy of different methods used on the seven most populated categories of Reuters-21578. Table 5.14 depicts comparison results between the classifiers produced by the proposed algorithm against other well-known Text Classifiers. It should be noted that the results of the BCAR algorithm is reported in [63] while for MCAR the results were obtained via experiment. Comparison between mMCAR and Naive Bayes, Naive Bayes didn't have any result better than mMCAR. The mMCAR has better in five data sets, when kNN achieved better results in two data sets (*Crude* and *Interest*). The mMCAR better in five data sets, when SVM has achieved better results in two data sets (*Crude* and *Interest*). The mMCAR has achieved better results in six data sets when CBA achieved better result one data set (*Interest*). the mMCAR achieved better in five data set when, the MCAR has achieved better results in three data set (*Crude*, *Earn* and *Trade*). The mMCAR better in five data set the BCAR has achieved better results in two data sets (*Crude* and *Interest*). As a result, mMCAR algorithm achieved good accuracy on overall data sets.

Table 5. 13
 Classification Accuracy on Reuters Data Sets

Category/Algorithm	Naïve Bayes	kNN	SVM	CBA	MCAR	BCAR	mMCAR
Acq	91.5	92	95.2	89.9	90.2	97.8	98.4
Crude	81	85.7	88.7	77	88.1	88.1	81.7
Earn	95.9	97.3	98.4	89.2	99.8	97.4	98.4
Grain	72.5	88.2	91.8	72.1	95.3	86.5	98.5
Interest	58	74	75.4	70.1	41.6	83.5	59.2
Money-FX	62.9	78.2	75.4	72.4	74.3	84.4	93.2
Trade	50	77.4	77.3	69.7	96.2	89.8	95.9

Figure 5.3 shows the results of the proposed prediction method when our PRM pruning method is implemented. All our techniques outperformed the other traditional classification and association techniques and are slightly similar to BCAR.

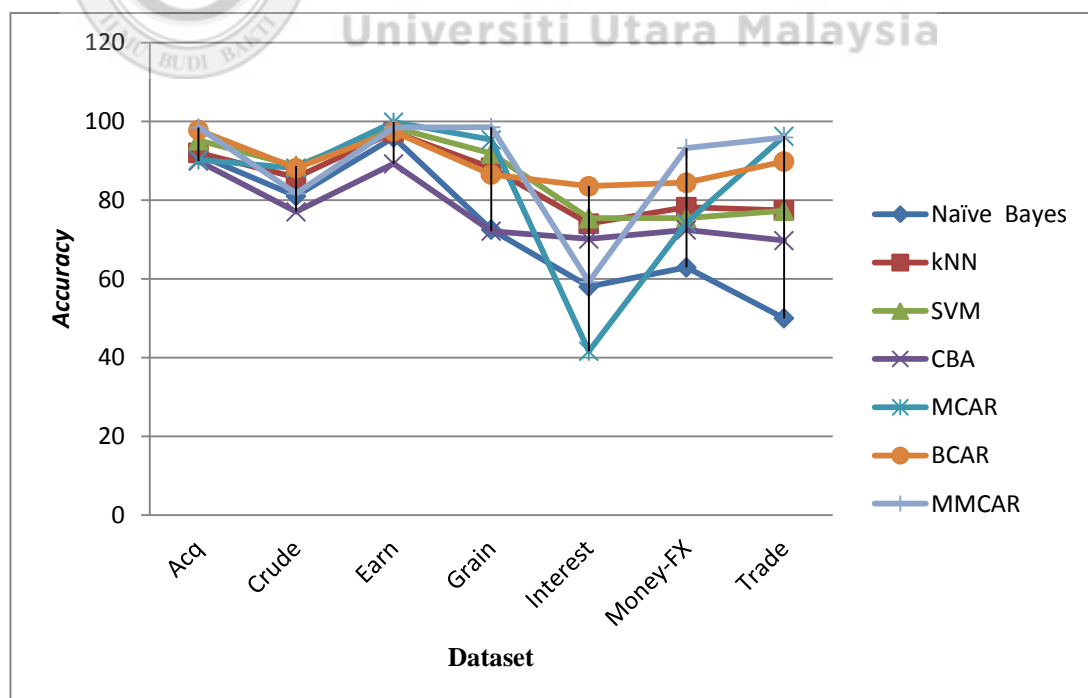


Figure 5.3: Classification Accuracy of Reuters Data Sets

Table 5.14
Classification Accuracy Between mMCAR and all Algorithms

Data set	Naïve					
	Bayes	mMCAR	kNN	mMCAR	SVM	mMCAR
Acq	91.5	98.4	92	98.4	95.2	98.4
Crude	81	81.7	85.7	81.7	88.7	81.7
Earn	95.9	98.4	97.3	98.4	98.4	98.4
Grain	72.5	98.5	88.2	98.5	91.8	98.5
Interest	58	59.2	74	59.2	75.4	59.2
Money-FX	62.9	93.2	78.2	93.2	75.4	93.2
Trade	50	95.9	77.4	95.9	77.3	95.9

Data set	CBA	mMCAR	MCAR	mMCAR	BCAR	mMCAR
Acq	89.9	98.4	90.2	98.4	97.8	98.4
Crude	77	81.7	88.1	81.7	88.1	81.7
Earn	89.2	98.4	99.8	98.4	97.4	98.4
Grain	72.1	98.5	95.3	98.5	86.5	98.5
Interest	70.1	59.2	41.6	59.2	83.5	59.2
Money-FX	72.4	93.2	74.3	93.2	84.4	93.2
Trade	69.7	95.9	96.2	95.9	89.8	95.9

Table 5.14 showed the accuracy performance of mMCAR comparing with other algorithm used in the experiment. According to the results, Naive Bayes didn't wins in any dataset compared with mMCAR. While, mMCAR has wins in five dataset, where, kNN has win in two dataset as a more accurate. Furthermore, mMCAR win with four dataset, while SVM has win in two dataset. mMCAR win in six dataset, when CBA win in one dataset more accurate. mMCAR has win in four dataset, while MCAR has win in three dataset more accurate. Finally, mMCAR has win in five dataset, when BCAR get two dataset more accurate.

5.4.2 Number of rule for Reuters Data Set

Table 5.15 displays the Reuters text collection derived into the number of rules when used on different pruning approaches. mMCAR algorithm using PRM approach as well as No pruning, Database Coverage and Lazy. Comparison between mMCAR and No pruning, No pruning didn't have any result better than mMCAR. In other hand mMCAR achieved better in five

data sets, when Database Coverage achieved better results in two data set, (*Interest* and *Money-FX*). mMCCAR achieved better in six data sets when, Lazy achieved better results in one data set, (*Money-FX*). mMCCAR decrease number of rules in the most cases.

*Table 5.15
Number of Rules Using Pruning Approach*

Category Name	No pruning	Database Coverage (CBA, MCAR)	LAZY (BCAR)	PRM mMCCAR
Acq	80	27	40	16
Crude	8	4	6	4
Earn	172	17	55	16
Grain	5	5	5	5
Interest	4	2	4	3
Money-FX	23	12	15	16
Trade	9	6	8	6

In particular, for all classification data sets we considered, Figure 5.4 mMCCAR algorithm using PRM produces fewer rules than the other methods. One of the main reasons for generating a large number of rules is storing rules that cover at least one training document regardless whether the rules classify training document correctly. For example, the number of rules generated without pruning method on "Acq" data set is 80, whereas rules are generated using PRM is 16; these 64 rules may decrease the accuracy and the classification time may increase. Result show that PRM reduced the number of rule in 5 dataset as a total reduces the number of rule for all data.

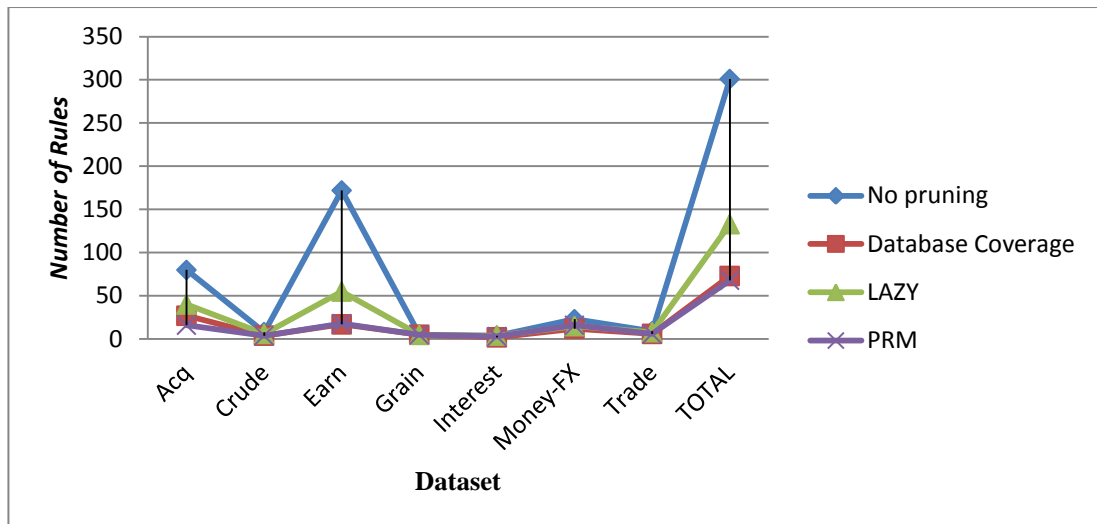


Figure 5.4 Number of Rules Using Pruning Approaches

Furthermore, Table 5.16 showed the result for the number of rules of mMCAR comparing with other algorithm used in the experiment. The algorithm of No pruning didn't win in any dataset as less number of rules than mMCAR. While, mMCAR also win in two data sets, when Database Coverage has win in two dataset. Lastly, mMCAR has win in five data sets, when Lazy has win in one dataset as a less number of rules.

Table 5.16
Number of Rules Between mMCAR and All Algorithms

Category Name	No pruning	PRM mMCAR	Database Coverage (CBA, MCAR)	PRM mMCAR	LAZY (BCAR)	PRM mMCAR
Acq	80	16	27	16	40	16
Crude	8	4	4	4	6	4
Earn	172	16	17	16	55	16
Grain	5	5	5	5	5	5
Interest	4	3	2	3	4	3
Money-FX	23	16	12	16	15	16
Trade	9	6	6	6	8	6

5.4.3 The Win-Loss-Tie Record

Table 5.17 illustrates the Win/Lose/Tie for accuracy record of the recommended algorithm mMCAR out of all considered algorithms from Table 5.13. The three values (Win/Lose/Tie) record are the number of data sets for which a method obtains higher, lower or equal accuracy respectively as compared to an alternative method. The Win/Lose/Tie record of the proposed algorithm are listed in Table 5.17 against the selected competitors for average classification rates on the datasets. Won-Lost-Tie record for accuracy of mMCAR against Naïve Bayes, K-NN, SVM, CBA, MCAR and BCAR are 7-0-0, 5-2-0, 4-2-1, 6-1-0, 4-3-0 and 5-2-0 respectively. Table 5.15 on the other hand shows the won-loss-tied record for the number of rules in mMCAR against all selected algorithms. We can conclude from Table 5.18 that the Win/Lose/Tie record of pruning methods mMCAR and PRM against no pruning, database converge and lazy are 6-0-1, 2-2-3, and 5-1-1 respectively. Therefore, the mMCAR algorithm are proved to be better against all selected algorithms on numerous datasets.

*Table 5.17
Results on Win/Lose/Tie for Accuracy*

Category/Algorithm	Naïve Bayes	NN	SVM	CBA	MCAR	BCAR
mMCAR	7-0-0	5-2-0	4-2-1	6-1-0	4-3-0	5-2-0

*Table 5.18
Results on Win/Lose/Tie for Number of Rule*

Category/Algorithm	No Pruning	Database Coverage (CBA, MCAR)	LAZY (BCAR)
mMCAR PRM	6-0-1	2-2-3	5-1-1

The impact between the number of rules and the accuracy is expounded. The result illustrates that the PRM obtains the smallest number of rules in total. PRM considers partial matching between the training instance and the rule during the evaluation step and this is the actual reason of having small number of rules generated. This means that the rule can cover a large number of training instances than the full matching procedure. As a result, number of rules would be less that covers more training instances unlike LAZY and database coverage which considers a rule to be significant only that rule can cover the training case with full matching between the training-case attribute values and the body of the case.

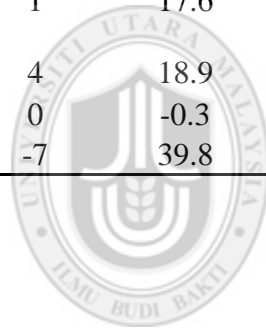
5.4.4 Compression Variation between AC algorithms

The variation between AC algorithms CBA, MCAR, BCAR and mMCAR are illustrated in Table 5.19 below along the variation for all data sets and total variations.

Table 5.19

The variation of Reuter's Data Set Between AC Algorithms

Data set	n.rule mMCAR vs CBA	Accuracy mMCAR vs CBA	n.rule mMCAR vs MCAR	Accuracy mMCAR vs MCAR	n.rule mMCAR vs BCAR	Accuracy mMCAR vs BCAR	n.rule MCAR vs CBA	Accuracy MCAR vs CBA	n.rule BCAR vs CBA	Accuracy BCAR vs CBA	n.rule BCAR vs MCAR	Accuracy BCAR vs MCAR
Acq	-11	8.5	-11	8.2	-24	0.6	0	0.3	13	7.9	13	7.6
Crude	0	4.7	0	-6.4	-2	-6.4	0	11.1	2	11.1	2	0
Earn	-1	9.2	-1	-1.4	-39	1	0	10.6	38	8.2	38	-2.4
Grain	0	26.4	0	3.2	0	12	0	23.2	0	14.4	0	-8.8
Interest	1	-10.9	1	17.6	-1	-24.3	0	-28.5	2	13.4	2	41.9
Money-FX	4	20.8	4	18.9	1	8.8	0	1.9	3	12	3	10.1
Trade	0	26.2	0	-0.3	-2	6.1	0	26.5	2	20.1	2	-6.4
total	-7	84.9	-7	39.8	-67	-2.2	0	45.1	60	87.1	60	42



UUM
Universiti Utara Malaysia

Table 5.19 illustrates the results of the variations between AC algorithm in the experiment. The first column in the Table above shows the variations between CBA and mMCAR in number of rules. The total result show that mMCAR decrease 7 rules. The second column of above Table depicts the variations in the accuracy between CBA and mMCAR and the total result show mMCAR is better in terms of accuracy and increases the accuracy 84.9 as a total. The variations in number of rules between mMCAR and MCAR are shown in third column of above Table. The total result reveal that mMCAR decrease 7 rules. The variation in the accuracy between mMCAR and MCAR are given in fourth column of the above Table and the total result show mMCAR increase the accuracy 39.8 in total. The variations between mMCAR and BCAR in number of rule are given in the fifth column of the Table and the total result show that mMCAR decrease 67 rules. The variation in the accuracy between mMCAR and BCAR are given in sixth column of the above Table which shows that mMCAR is better in accuracy by -2.2 as a total. The seventh column shows the variation between CBA and the MCAR in number of rule which shows that MCAR decrease 0 rules while the eighth column show the variation in the accuracy between CBA and MCAR and the total result show that MCAR increase the accuracy 45.1 in total. The variations between BCAR and CBA in number of rules are shown in the ninth column of the above Table and the result show that BCAR increase 60 rules. The tenth column of the Table expresses the variation in the accuracy between BCAR and CBA and the total result reveal that BCAR increase the accuracy 87.1 in total. The variation between BCAR and MCAR in number of rule are illustrated in eleventh column of the Table which shows that that BCAR increases 60 rules. The variation in the accuracy between BCAR and MCAR are given in twelfth column of the Table above which shows that BCAR increase the

accuracy by 42 in total. Conclusively, mMCAR as compared to CBA, MCAR and BCAR decreases the number of rules as a total in 7-7-67 respectively. While in terms of accuracy, mMCAR can increase the accuracy in total for CBA and MCAR 84.9 and 39.8 respectively. mMCAR got slightly decrease comparing to BCAR in -2.2 when decrease number of rule 67. The result reveals that our objective is achieved as to have a higher accuracy with reduced number of rules.

5.4.5 Training and Testing Time for Reuter's Data Set

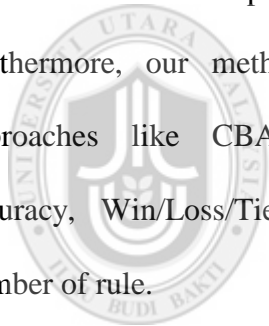
This section show the time taken for AC algorithm (CBA, MCAR, BCAR and mMCAR) to building the classifier on the the training and testing time for Reuter's data set in minsupp 2% and minconf 50%. Table 5.20 show the CPU time taken for mMCAR, is 21.44 seconds for training data and 9.83 seconds for testing data. The runtime revealed that mMCAR is faster than Naive bayes, kNN, SVM, CBA, MCAR and BCAR in the training and testing time. The vertical and horizontal intersection method that mMCAR employed to find the rules and avoiding going over the data multiple time during building the classifier, are responsible for the runtime advantage.

*Table 5.20
Training and Testing Time for Reuter's Data Sets*

Method	Training time (second)	Testing time (second)
Naïve Bayes	34.64	16.38
kNN	32.36	14.92
SVM	29.73	12.27
CBA	27.37	11.21
MCAR	36.22	17.15
BCAR	42.28	28.3
mMCAR	21.44	9.38

5.5 Summary

In this chapter, a new classification based association rule algorithm called mMCAR has been proposed. The First experiment was to investigate structured data UCI and the results showed highly competitive when compared with other algorithms such as RIPPER, C4.5, and MCAR in terms of prediction accuracy, Win/Loss/Tie, and number of rule. In other hand, compression variation between AC algorithms mMCAR can reduced the rule and get a comparative accuracy. The second experimental, for unstructured data Reuters-21578 the results is highly competitive when compared with traditional classification algorithms such as SVM, KNN, and Bayes in terms of prediction accuracy, Win/Loss/Tie, and number of rule. Furthermore, our method get a good result if compared with popular AC approaches like CBA, MCAR and BCAR with regards to prediction accuracy, Win/Loss/Tie, compression variation between AC algorithms and number of rule.



UUM
Universiti Utara Malaysia

CHAPTER SIX

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This study investigated the problem of generating rules with single label multi class using an AC approach for structured data (UCI) and unstructured data (Reuters-21578). This study proposed a new AC algorithm (mMCAR) which applies three new methods, namely, rule discovery, pruning and prediction. The contributions of this study is summarised in this section.

6.2 Rule Discovery Algorithm that Reduces Computational Time

This study presents a rule discovery algorithm (refer to section 4.3) that uses vertical horizontal format representation in where each itemset has a tid-list consisting the row numbers in which the item has occurred in the database. The algorithms that utilised vertical and horizontal format have shown to be more effective and often better than horizontal techniques or vertical techniques and our CPU time results support that (Table 5.3, 5.7). The proposed method goes over the training data set only once to count the frequencies of ruleitems. However, any items that did not pass the MinSupp are removed. The MinConf is then calculated and in the event an item did not pass the MinConf threshold, it will be removed. This representation enables mMCAR to intersect the tid lists of frequent 1-ruleitem to extract candidate ruleitems in which the cardinality of any intersection operation between 2 and 1-ruleitems give the frequency (support value) of the resulting 2-candidate ruleitems. The same process is applied on frequent 2-ruleitems to discover candidate 3-rulesitems and so on.

6.3 Rule Pruning Algorithm that Reduces the Number of Classification Rules

The large number of rules is main issue in this research especially if the training data set is large. We want to have a small number of the most powerful rules. We present in this thesis new rule pruning method called Partly Rule Match (PRM) and this is presented in Figure 4.4. Experimental results showed that the proposed rule pruning methods improve the accuracy of output system and reduced the rule.

The results show PRM get the smallest number of rule in average from the two experiments in Chapter Five. The main reason for the less number of rules generated for the PRM algorithm is that during the rule evaluation step, PRM considers partly matching between the rule and the training instance. This makes the rule covers a larger number of training instances than the procedure that requires full matching. Consequently, there will be less number of rules covering more training instances unlike other methods which considers a rule significant if it covers the training case with full matching between its body and the training case attribute values.

6.4 Rule Prediction Algorithm that Improves Accuracy

Prediction is one of the important steps that play a major role to increase the accuracy for the system. The challenge here is how to make use of the set of significant rules generated after the rule pruning in order to give a good prediction.

In this thesis we present new prediction method, which is the Joint Confidence Support Class Prediction (JCSCP). The JCSCP (presented as Figure 4.5) splits rules into groups based on the class value and for each group it computes its weight. The weighted average counts for every rule (support and confidence) and choose the predicted class. Experimental results showed that our prediction methods outperformed other classification methods in the two experiment, namely for structured and unstructured data.

6.5 Future Work

6.5.1 Multi-label in Text Classification

We intend to extend our work to develop multi-label Association algorithms using vertical layout to handle TC problem by extracting very useful knowledge missed by current approaches. Consider for example, a document which has two class labels “Health” and “sport”, and assume that the document is associated 40 times with the “Health” label and 38 times with the “sport” label, and the number of times the document appears in the training data is 78. A traditional AC algorithm extracts only the rule associated with the most obvious label, i.e. “Health”, for the fact that it has the largest in occurrence, and even ignores the other potential rule. However, it is of benefit to extract the other rules, since they at times bring up useful information with a large representation in the database. Meaning that the ignored rule may also take a role in prediction and may be very of importance to the decision maker.

6.5.2 Discretisation

The first step in TC i.e. pre-processing including remove numbers, stop word and feature selection. But the text may contain numbers (continues data) that have significant value like the Independence Day and birth dates. One possible future direction is to treat text with continuous data. For continuous attributes, the Multi-interval discretisation technique of [113] can be implemented within an AC algorithm. The process of discretising continuous attributes is briefly summarise by the researcher from [113]. “First, the training cases for each continuous attribute are sorted in ascending order and the class values associated with each case is given.

The next step is to place break points whenever the class value changes and to calculate the information gain[120] for each possible break point. The information gain represents the amount of information required to specify values of the classes given a breaking point”. Finally, the break point that minimises the information gain over all possible breaking points is selected and the algorithm is invoked again on the lower range of that attribute.

6.5.3 Pre-Pruning

There are three phases for the traditional algorithms of AC, namely, prediction, classifier construction, and rule generation. Rule generation employs the association rule mining technique to search for the frequent patterns containing the classification rules. Building the classifier phase removes the redundant rules, and organises the significant rules. Finally, the unlabeled data are classified in the third step. Experiments conducted in AC

such as CBA [43], CMAR [30], and BCAR[63], state that the AC methods share the fact that even with the present post pruning methods such as the database coverage, the number of rules in the classifier is still large. This increases the time cost when predicting test cases.

To the best of author's knowledge, there are some initial attempts to tackle the problem of searching space in AC in order to cut down the number of candidate rules [38, 43]. Thus, reducing the searching space before generating rules is an important future direction. In other words, we want to limit the number of candidate or frequent ruleitems before the rules get generated.

6.6 Summary

This study in association classification text mining, new AC algorithm (mMCAR) was proposed, this chapter contain three contribution these contribution are, rule discovery algorithm to reduce the disjoin items this will reduce the computational time, new rule pruning method called Partly Rule Match (PRM) to reduce the number of rules and new rule prediction method Joint Confidence Support Class Prediction (JCSCP) to enhance the accuracy, as well as in this chapter also mentioned the future work for this study.

REFERENCE

- [1] U. Fayyad, *et al.*, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, p. 37, 1996.
- [2] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*: Morgan Kaufmann Pub, 2005.
- [3] W. Han, *et al.*, "Research on the Problem Model of GUI based on Knowledge Discovery in Database," in *2013 International Conference on Software Engineering and Computer Science*, 2013.
- [4] A. Sharafi, *et al.*, "Knowledge Discovery in Databases on the Example of Engineering Change Management," in *Industrial Conference on Data Mining-Poster and Industry Proceedings*, 2010, pp. 9-16.
- [5] C. M. L. Antonie, "Associative classifiers: Improvements and potential," UNIVERSITY OF ALBERTA, 2009.
- [6] T. Dong, *et al.*, "The Research of kNN Text Categorization Algorithm Based on Eager Learning," in *Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on*, 2012, pp. 1120-1123.
- [7] A. C. Neocleous, *et al.*, "Artificial neural networks to investigate the importance and the sensitivity to various parameters used for the prediction of chromosomal abnormalities," in *Artificial Intelligence Applications and Innovations*, ed: Springer, 2012, pp. 46-55.
- [8] B. Sriram, *et al.*, "Short text classification in twitter to improve information filtering," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 2010, pp. 841-842.
- [9] R. Brause, "Medical analysis and diagnosis by neural networks," *Medical data analysis*, pp. 1-13, 2001.
- [10] G. J. Simon, *et al.*, "A simple statistical model and association rule filtering for classification," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 823-831.
- [11] W. Zhang, *et al.*, "A comparative study of TF* IDF, LSI and multi-words for text classification," *Expert Systems with Applications*, vol. 38, pp. 2758-2765, 2011.
- [12] F. Thabtah, *et al.*, "Arabic Text Mining Using Rule Based Classification," *Journal of Information & Knowledge Management*, vol. 11, 2012.
- [13] C. C. Aggarwal and C. Zhai, *Mining text data*: Springer, 2012.
- [14] S. Kiritchenko and S. Matwin, "Email classification with co-training," in *Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research*, 2011, pp. 301-312.
- [15] H. Dag, *et al.*, "Comparison of feature selection algorithms for medical data," in *Innovations in Intelligent Systems and Applications (INISTA), 2012 International Symposium on*, 2012, pp. 1-5.
- [16] A. James, *et al.*, "Research Directions in Database Architectures for the Internet of Things: A Communication of the First International Workshop on Database Architectures for the Internet of Things (DAIT 2009)," *Dataspace: The Final Frontier*, pp. 225-233, 2009.
- [17] Y. Zhu, *et al.*, "Font recognition based on global texture analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1192-1200, 2001.

- [18] J. R. Quinlan, *C4. 5: programs for machine learning*. San Mateo: Morgan Kaufmann, 1993.
- [19] X. Qi and B. D. Davison, "Web page classification: Features and algorithms," *ACM computing surveys (CSUR)*, vol. 41, p. 12, 2009.
- [20] Wu, Ho Chung, et al. "Interpreting tf-idf term weights as making relevance decisions." *ACM Transactions on Information Systems (TOIS)* 26.3 (2011)
- [21] G. Cormode and M. Hadjieleftheriou, "Methods for finding frequent items in data streams," *The VLDB Journal*, vol. 19, pp. 3-20, 2010.
- [22] D. Meretakakis and B. Wüthrich, "Extending naïve Bayes classifiers using long itemsets," 1999, pp. 165-174.
- [23] M. Henning, "The rise and fall of CORBA," *Communications of the ACM*, vol. 51, pp. 52-57, 2008.
- [24] L. Shi, et al., "Cross language text classification by model translation and semi-supervised learning," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 2010, pp. 1057-1067.
- [25] J. E. Gentle, et al., *Handbook of computational statistics: concepts and methods*: Springer, 2012.
- [26] E. Wiener, et al., "A neural network approach to topic spotting," in *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval*, 1995, pp. pp. 317-332.
- [27] K. Hornik, "Snowball: Snowball Stemmers," *Rpackage version 0.0-7*, URL <http://CRAN.R-project.org/package=Snowball>, 2009.
- [28] J. Duan, et al., "Scaling up the accuracy of Bayesian classifier based on frequent itemsets by m-estimate," in *Artificial Intelligence and Computational Intelligence*, ed: Springer, 2010, pp. 357-364.
- [29] G. Dong, et al., "CAEP: Classification by aggregating emerging patterns," Japan, 1999, pp. 737-737.
- [30] W. Li, et al., "CMAR: Accurate and efficient classification based on multiple class-association rules," in *Proceedings of the ICDM'01*, San Jose, CA, 2001, p. 369.
- [31] F. Thabtah, et al., "A New Classification Based on Association Algorithm," *Journal of Information & Knowledge Management*, vol. 9, p. 55 64, 2010.
- [32] J. Read, et al., "Classifier chains for multi-label classification," *Machine learning*, vol. 85, pp. 333-359, 2011.
- [33] K. Yu, et al., "Mining emerging patterns by streaming feature selection," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 60-68.
- [34] E. Baralis, et al., "A lazy approach to associative classification," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, pp. 156-171, 2008.
- [35] X. Y. J. Han, "CPAR: Classification based on predictive association rules," 2003, p. 331.
- [36] E. Baralis, et al., "On support thresholds in associative classification," in *Proceedings of the 2004 ACM Symposium on Applied Computing*, Nicosia, Cyprus, 2004, pp. 553-558.
- [37] F. Thabtah, et al., "MCAR: multi-class classification based on association rule," in *Proceeding of the 3rd IEEE International Conference on Computer Systems and Applications*, 2005, p. 33.
- [38] Z. Tang and Q. Liao, "A new class based associative classification algorithm," *IAENG International Journal of Applied Mathematics.-1998.-36: 2, IJAM.-. 136*, vol. 141, 2007.

- [39] Y. Yoon and G. G. Lee, "Text categorization based on boosting association rules," 2008, pp. 136-143.
- [40] D. Meretakis and B. Wüthrich, "Extending naïve Bayes classifiers using long itemsets," in *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, California, 1999, pp. 165-174.
- [41] R. Quinlan, "Data mining tools See5 and C5. 0," *Artificial Intelligence*, 2004.
- [42] E. Wiener, *et al.*, "A neural network approach to topic spotting," in *Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, 1995.
- [43] B. Liu, *et al.*, "Integrating classification and association rule mining," *Knowledge discovery and data mining*, pp. 80–86, 1998.
- [44] M. L. Antonie and O. Zaïane, "Mining positive and negative association rules: an approach for confined rules," *Knowledge Discovery in Databases: PKDD 2004*, pp. 27-38, 2004.
- [45] G. Kundu, *et al.*, "ACN: An associative classifier with negative rules," 2008, pp. 369-375.
- [46] F. A. Thabtah, *et al.*, "MMAC: A new multi-class, multi-label associative classification approach," 2004.
- [47] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the 20th International Conference on Very Large Data Bases* Santiago, Chile, 1994, pp. 487-499.
- [48] M. J. Zaki, *et al.*, "New algorithms for fast discovery of association rules," in *3rd KDD Conference* New York, 1997.
- [49] M. J. Zaki and K. Gouda, "Fast vertical mining using diffsets," in *Proceedings of the ninth ACM* Washington, D.C, 2003, pp. 326-335.
- [50] J. R. Quinlan, "Generating production rules from decision trees," in *Artificial Intelligence*, Milan, Italy., 1987, pp. 304-307.
- [51] G. Salton, "Automatic text processing: the transformation," *Analysis and Retrieval of Information by Computer*, vol. 14, p. 15, 1989.
- [52] L. T. Nguyen, *et al.*, "Classification based on association rules: A lattice-based approach," *Expert Systems with Applications*, vol. 39, pp. 11357-11366, 2012.
- [53] E. Baralis and P. Garza, "I-prune: Item selection for associative classification," *International Journal of Intelligent Systems*, vol. 27, pp. 279-299, 2012.
- [54] C.-H. Chen, *et al.*, "Improving the performance of association classifiers by rule prioritization," *Knowledge-Based Systems*, vol. 36, pp. 59-67, 2012.
- [55] M. G. Al Zamil and A. B. Can, "ROLEX-SP: Rules of lexical syntactic patterns for free text categorization," *Knowledge-Based Systems*, vol. 24, pp. 58-65, 2011.
- [56] Z. Zhou, *et al.*, "Association classification algorithm based on structure sequence in protein secondary structure prediction," *Expert Systems with Applications*, vol. 37, pp. 6381-6389, 2010.
- [57] J. Alcalá-Fdez, *et al.*, "A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning," *Fuzzy Systems, IEEE Transactions on*, vol. 19, pp. 857-872, 2011.
- [58] Z. Zhang and R. S. Blum, "A categorization of multiscale-decomposition-based image fusion schemes with a performance study for a digital camera application," *Proceedings of the IEEE*, vol. 87, pp. 1315-1326, 1999.

- [59] B. Starfield, *et al.*, "Ambulatory care groups: a categorization of diagnoses for research and management," *Health Services Research*, vol. 26, p. 53, 1991.
- [60] P. C. Austin, *et al.*, "Comparative ability of comorbidity classification methods for administrative data to predict outcomes in patients with chronic obstructive pulmonary disease," *Annals of epidemiology*, 2012.
- [61] H. Shatkay, *et al.*, "Integrating image data into biomedical text categorization," *Bioinformatics*, vol. 22, p. e446, 2006.
- [62] F. Thabtah, *et al.*, "MCAR: multi-class classification based on association rule," 2005, p. 33.
- [63] A. Chang, *et al.*, "An Integer Optimization Approach to Associative Classification," in *Advances in Neural Information Processing Systems*, 2012, pp. 269-277.
- [64] M. L. G. a. t. U. o. Waikato. stemmer. Available: http://www.cs.waikato.ac.nz/~ml/weka/index_downloading.html
- [65] F. THABTAH and S. HAMMOUD, "MR-ARM: A MAP-REDUCE ASSOCIATION RULE MINING FRAMEWORK," *Parallel Processing Letters*, vol. 23, 2013.
- [66] S. Z. H. Zaidi, *et al.*, "Distributed data mining from heterogeneous healthcare data repositories: towards an intelligent agent-based framework," 2002, pp. 339-342.
- [67] I. Yeh, *et al.*, "Applications of web mining for marketing of online bookstores," *Expert Systems with Applications*, vol. 36, pp. 11249-11256, 2009.
- [68] C. C. Aggarwal, "Collaborative crawling: Mining user experiences for topical resource discovery," 2002, pp. 423-428.
- [69] D. D. Lewis. (2004, Reuters-21578. Available: <http://www.daviddlewis.com/resources/testcollections/reuters21578/>
- [70] G. Chen, *et al.*, "A new approach to classification based on association rule mining," *Decision Support Systems*, vol. 42, pp. 674-689, 2006.
- [71] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, pp. 1-13, 2007.
- [72] J. Balcázar, "Minimum-size bases of association rules," *Machine Learning and Knowledge Discovery in Databases*, vol. 5211, pp. 86-101, 2008.
- [73] Q. Niu, *et al.*, "Association Classification Based on Compactness of Rules," in *Second International Workshop on Knowledge Discovery and Data Mining*, 2009, pp. 245-247.
- [74] H. Ishibuchi, *et al.*, "Prescreening of candidate rules using association rule mining and Pareto-optimality in genetic rule selection," 2007, pp. 509-516.
- [75] J. Han, *et al.*, *Data mining: concepts and techniques*: Morgan Kaufmann Pub, 2011.
- [76] C. Merz and P. Murphy, "UCI repository of machine learning databases, 1996," *FTP from ics. uci. edu in the directory pub/machine-learning-databases*.
- [77] D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," *Machine Learning: ECML-98*, pp. 4-15, 1998.
- [78] L. Alvim, *et al.*, "Sentiment of financial news: a natural language processing approach," in *1st Workshop on Natural Language Processing Tools Applied to Discourse Analysis in Psychology*, Buenos Aires, 2010.

- [79] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," *Machine Learning: ECML-98*, pp. 137-142, 1998.
- [80] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," Nashville, TN, 1997, pp. 412-420.
- [81] F. Sebastiani, "A tutorial on automated text categorisation," in *1st Argentinian Symposium on Artificial Intelligence*, 1999, pp. 7-35.
- [82] T. Tokunaga and I. Makoto, "Text categorization based on weighted inverse document frequency," in *the Special Interest Groups and Information Process Society of Japan (SIG-IPSI)*, Tokyo, Japan, 1994.
- [83] C. Deisy, *et al.*, "A novel term weighting scheme MIDF for Text Categorization," *Journal of Engineering Science and Technology*, vol. 5, pp. 94-107, 2010.
- [84] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval* vol. 463: ACM press New York, 1999.
- [85] A. R. Pal, *et al.*, "An Approach To Automatic Text Summarization Using Simplified Lesk Algorithm And Wordnet," *International Journal of Control Theory & Computer Modeling*, vol. 3, 2013.
- [86] C. J. Rijsbergen, "Information retrieval," *A statistical interpretation of term specificity and its application in retrieval*," *Journal of documentation*, vol. 28, 1979.
- [87] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, pp. 11-21, 1972.
- [88] F. Thabtah and H. Abdel-jaber, "A Comparative Study using Vector Space Model with K-Nearest Neighbor on Text Categorization Data," in *Proceedings of the 2007 International Conference of Data Mining and Knowledge Engineering*, London, UK, 2007.
- [89] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, pp. 81-106, 1986.
- [90] G. W. Snedecor and W. Cochran, "Statistical methods," *Statistical methods*, 1989.
- [91] T. M. Mitchell, *Machine learning*. WCB/McGraw-Hill, New York, New York: Artificial Neural Networks, 1997.
- [92] V. N. Vapnik, *The nature of statistical learning theory*. New York: Springer Verlag, 2000.
- [93] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, 1999, pp. 42-49.
- [94] X. Zhang and H. Huang, "An improved KNN text categorization algorithm by adopting cluster technology," *Pattern Recognit Artif Intell*, vol. 22, pp. 936-940, 2009.
- [95] B. Xu, *et al.*, "An Improved Random Forest Classifier for Text Categorization," *Journal of Computers*, vol. 7, pp. 2913-2920, 2012.
- [96] K. Tzeras and S. Hartmann, "Automatic indexing based on Bayesian inference networks," in *Proceedings of the 16th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*, 1993, pp. 22-35.
- [97] S. Jiang, *et al.*, "An improved k-nearest-neighbor algorithm for text categorization," *Expert Systems with Applications*, vol. 39, pp. 1503-1509, 2012.

- [98] P. Cunningham and S. J. Delany, "k-Nearest neighbour classifiers," *Multiple Classifier Systems*, pp. 1-17, 2007.
- [99] M. F. Othman and T. M. S. Yau, "Comparison of different classification techniques using WEKA for breast cancer," in *IFMBE Proceedings Springer*, Malaysia, 2007, pp. 520-523.
- [100] G. A. Wa'el Musa Hadi and F. Thabtah, "VSMs with K-Nearest Neighbour to Categorise Arabic Text Data," in *Proceedings of the European Simulation and Modelling Conference*, Le Havre, France, 2008.
- [101] R. O. Duda and P. E. Hart, "Pattern classification and scene analysis," *A Wiley-Interscience Publication, New York: Wiley, 1973*, vol. 1, 1973.
- [102] X. Ma, *et al.*, "Combining Naive Bayes and Tri-gram Language Model for Spam Filtering," in *Knowledge Engineering and Management*, ed: Springer, 2012, pp. 509-520.
- [103] M. Elmarhoumy, *et al.*, "A new modified centroid classifier approach for automatic text classification," *IEEJ Transactions on Electrical and Electronic Engineering*, 2013.
- [104] F. Denis, *et al.*, "Efficient learning of Naive Bayes classifiers under class-conditional classification noise," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 265-272.
- [105] R. E. Schapire, *et al.*, "Boosting and Rocchio applied to text filtering," in *ACM*, 1998, pp. 215-223.
- [106] D. D. Jensen and P. R. Cohen, "Multiple comparisons in induction algorithms," *Machine learning*, vol. 38, pp. 309-338, 2000.
- [107] Z. Wang, *et al.*, "A Multiclass SVM Method via Probabilistic Error-Correcting Output Codes," in *Internet Technology and Applications, 2010 International Conference on*, 2010, pp. 1-4.
- [108] P. Y. Pawar and S. Gawande, "A Comparative Study on Different Types of Approaches to Text Categorization," *International Journal of Machine Learning and Computing*, vol. 2, 2011.
- [109] T. Kohonen and P. Somervuo, "Self-organizing maps of symbol strings," *Neurocomputing*, vol. 21, pp. 19-30, 1998.
- [110] T. S. Lim, *et al.*, "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms," *Machine learning*, vol. 40, pp. 203-228, 2000.
- [111] F. Odeh and N. Al-Najdawi, "ACNB: Associative Classification Mining Based on Naïve Bayesian Method," *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 8, pp. 23-35, 2013.
- [112] X. Li, *et al.*, "ACCF: Associative Classification Based on Closed Frequent Itemsets," 2008, pp. 380-384.
- [113] U. Fayyad and K. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," 1993.
- [114] W. Li, "Classification based on multiple association rules," Citeseer, 2001.
- [115] T. Qian, *et al.*, "2-ps based associative text classification," *Data Warehousing and Knowledge Discovery*, pp. 378-387, 2005.
- [116] M. J. Zaki and C. J. Hsiao, "CHARM: An efficient algorithm for closed itemset mining," 2002.
- [117] R. E. Schapire, "Using output codes to boost multiclass learning problems," in *Machine Learning*, 1997, pp. 313-321.

- [118] Q. Niu, *et al.*, "Association Classification Based on Compactness of Rules," in *Second International Workshop on Knowledge Discovery and Data Mining.*, 2009, pp. 245-247.
- [119] J. Han, *et al.*, "Mining frequent patterns without candidate generation," 2000, pp. 1-12.
- [120] F. A. Thabtah and P. I. Cowling, "A greedy classification algorithm based on association rule," *Applied Soft Computing*, vol. 7, pp. 1102-1111, 2007.
- [121] B. Cule and B. Goethals, "Mining association rules in long sequences," in *Advances in Knowledge Discovery and Data Mining*, ed: Springer, 2010, pp. 300-309.
- [122] O. R. Zaïane and M. L. Antonie, "Classifying text documents by associating terms with text categories," in *Australasian conference on database technologies*, Melbourne, Australia, 2003, pp. 215-222.
- [123] I. H. Witten, *et al.*, *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques*: Morgan Kaufmann, 2011.
- [124] J. Jabez Christopher, "A Statistical Approach for Associative Classification," *European Journal of Scientific Research*, vol. 58, pp. 140-147, 2011.
- [125] S. Maffeis and D. C. Schmidt, "Constructing reliable distributed communication systems with CORBA," *Communications Magazine, IEEE*, vol. 35, pp. 56-60, 1997.
- [126] F. Thabtah, *et al.*, "Rule Pruning Methods in Associative Classification Text Mining," *Journal of Intelligent Computing Volume*, vol. 1, p. 1, 2010.
- [127] S. Sangsuriyun, *et al.*, "Hierarchical Multi-label Associative Classification (HMAC) using negative rules," in *IEEE International Conference*, Bangkok, 2010, pp. 919-924.
- [128] P. Clark and R. Boswell, "Rule induction with CN2: Some recent improvements," in *Machine Learning*, Berlin, 1991, pp. 151-163.
- [129] M. L. Antonie and O. R. Zaïane, "Text document categorization by term association," 2002.
- [130] M. L. Antonie, *et al.*, "Associative classifiers for medical images," *Mining Multimedia and Complex Data*, pp. 68-83, 2003.
- [131] W. C. Chen, *et al.*, "Increasing the effectiveness of associative classification in terms of class imbalance by using a novel pruning algorithm," *Expert Systems with Applications*, 2012.
- [132] E. Baralis and J. Widom, "An algebraic approach to static analysis of active database rules," *ACM Transactions on Database Systems (TODS)*, vol. 25, pp. 269-332, 2000.
- [133] F. Thabtah, *et al.*, "MCAR: multi-class classification based on association rule," in *Proceeding of the 3rd IEEE International Conference on Computer Systems and Applications* Cairo, Egypt., 2005, pp. 1-7.
- [134] M. L. Antonie and O. R. Zaïane, "Text document categorization by term association," 2002, pp. 19-26.
- [135] F. Thabtah, *et al.*, "Comparison of rule based classification techniques for the Arabic textual data," 2011, pp. 105-111.
- [136] T. D. Do, *et al.*, "Prediction confidence for associative classification," Singapore 2005, pp. 1993-1998.
- [137] M. Hall, *et al.*, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, pp. 10-18, 2009.

- [138] A. A. Freitas, "Understanding the crucial differences between classification and discovery of association rules: a position paper," *ACM SIGKDD Explorations Newsletter*, vol. 2, pp. 65-69, 2000.
- [139] M. Kantardzic and A. Badia, "Efficient Implementation of Strong Negative Association Rules," 2003, pp. 23-24.
- [140] R. Feldman and J. Sanger, *The text mining handbook: advanced approaches in analyzing unstructured data*: Cambridge Univ Pr, 2007.
- [141] B. Baharudin, *et al.*, "A review of machine learning algorithms for text-documents classification," *Journal of Advances in Information Technology*, vol. 1, pp. 4-20, 2010.
- [142] M. Lan, *et al.*, "Supervised and traditional term weighting methods for automatic text categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 721-735, 2009.
- [143] S. M. Weiss, *Text mining: predictive methods for analyzing unstructured information*: Springer-Verlag New York Inc, 2005.
- [144] M. J. Zaki and K. Gouda, "Fast vertical mining using diffsets," in *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, D.C, 2003, pp. 326-335.
- [145] Y. Yusof and M. H. Refai, "MMCAR: Modified multi-class classification based on association rule," in *Information Retrieval & Knowledge Management (CAMP), 2012 International Conference on*, 2012, pp. 6-11.
- [146] W. W. Cohen, "Fast effective rule induction," 1995, pp. 115-123.
- [147] B. Atmani and B. Beldjilali, "Knowledge discovery in database: Induction graph and cellular automaton," *Computing and Informatics*, vol. 26, pp. 171-197, 2012.
- [148] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, pp. 429-449, 2002.
- [149] Credé, Marcus, *et al.* "An evaluation of the consequences of using short measures of the Big Five personality traits." *Journal of personality and social psychology* 102.4 (2012): 874.
- [150] Quinlan, J. Ross. *C4. 5: programs for machine learning*. Elsevier, 2014.

APPENDIX A

Part of structure data UCI Dataset

@attribute 1

@attribute 2

@attribute 3

@attribute 4

@attribute 5

@attribute 6

@attribute 7

@attribute 8

@attribute 9

@attribute 10

@attribute 11

@attribute class

@data

x,male,notang,fal,norm,147\5+,fal,- 1\7,up,0\5+,rev,buff
x,fem,notang,fal,abn,- 147\5,fal,- 1\7,flat,- 0\5,norm,buff
y,fem,abnang,fal,hyp,147\5+,fal,- 1\7,flat,- 0\5,norm,buff
y,fem,notang,fal,hyp,147\5+,true,- 1\7,up,- 0\5,norm,buff
y,fem,asympt,fal,norm,147\5+,fal,- 1\7,up,- 0\5,norm,buff
x,fem,asympt,fal,norm,147\5+,fal,- 1\7,up,- 0\5,norm,buff
x,fem,asympt,fal,norm,- 147\5,fal,1\7+,flat,0\5+,norm,buff
y,fem,notang,true,norm,147\5+,fal,- 1\7,up,- 0\5,norm,buff
y,male,notang,fal,hyp,147\5+,fal,- 1\7,up,- 0\5,norm,buff
y,male,notang,fal,norm,- 147\5,true,- 1\7,flat,- 0\5,norm,buff
y,fem,asympt,fal,hyp,147\5+,true,- 1\7,flat,- 0\5,norm,buff
y,fem,abnang,fal,norm,147\5+,fal,- 1\7,up,- 0\5,norm,buff
y,male,asympt,fal,hyp,- 147\5,true,- 1\7,up,- 0\5,rev,buff
y,fem,notang,fal,hyp,147\5+,fal,- 1\7,up,- 0\5,norm,buff
x,male,abnang,fal,norm,147\5+,fal,- 1\7,up,- 0\5,norm,buff
y,male,asympt,true,norm,- 147\5,fal,- 1\7,up,0\5+,rev,buff
x,fem,notang,fal,hyp,- 147\5,fal,- 1\7,flat,- 0\5,norm,buff
y,male,asympt,fal,hyp,147\5+,fal,- 1\7,up,0\5+,norm,sick
x,male,notang,true,hyp,- 147\5,true,- 1\7,flat,0\5+,fix,sick
y,male,asympt,fal,norm,- 147\5,true,1\7+,flat,0\5+,rev,sick
y,male,asympt,fal,hyp,- 147\5,true,1\7+,flat,0\5+,norm,sick
x,fem,asympt,fal,norm,147\5+,true,1\7+,flat,0\5+,norm,sick
y,male,asympt,fal,norm,- 147\5,true,1\7+,flat,0\5+,rev,sick
x,male,asympt,true,norm,147\5+,fal,- 1\7,flat,0\5+,rev,sick
x,male,asympt,fal,abn,- 147\5,fal,1\7+,down,0\5+,fix,sick
y,male,asympt,true,hyp,- 147\5,true,- 1\7,flat,?,rev,sick
x,male,asympt,fal,norm,- 147\5,fal,- 1\7,flat,- 0\5,norm,sick
y,male,notang,fal,norm,- 147\5,fal,1\7+,flat,0\5+,rev,sick
x,male,asympt,fal,hyp,- 147\5,fal,1\7+,flat,0\5+,fix,sick
x,male,angina,fal,hyp,147\5+,fal,- 1\7,flat,- 0\5,rev,sick
x,male,asympt,fal,hyp,- 147\5,fal,1\7+,flat,0\5+,rev,sick

x,fem,notang,fal,norm,147\5+,fal,- 1\7,up,- 0\5,norm,buff

Part of structure data UCI dataset

@attribute 1
@attribute 2
@attribute 3
@attribute 4
@attribute class
@data
6\15+, 2\95-3\35, 4\75+, 1\75+, Iris-virginica
5\55-6\15, - 2\95, 4\75+, 0\8-1\75, Iris-virginica
6\15+, - 2\95, 4\75+, 1\75+, Iris-virginica
6\15+, 3\35+, 4\75+, 1\75+, Iris-virginica
6\15+, 2\95-3\35, 4\75+, 0\8-1\75, Iris-virginica
- 5\55, - 2\95, 2\45-4\75, 0\8-1\75, Iris-versicolor
5\55-6\15, - 2\95, 4\75+, 0\8-1\75, Iris-versicolor
- 5\55, - 2\95, 2\45-4\75, 0\8-1\75, Iris-versicolor
5\55-6\15, - 2\95, 2\45-4\75, 0\8-1\75, Iris-versicolor
5\55-6\15, - 2\95, 2\45-4\75, 0\8-1\75, Iris-versicolor
- 5\55, 2\95-3\35, - 2\45, - 0\8, Iris-setosa
- 5\55, 2\95-3\35, - 2\45, - 0\8, Iris-setosa
- 5\55, 3\35+, - 2\45, - 0\8, Iris-setosa
- 5\55, 2\95-3\35, - 2\45, - 0\8, Iris-setosa
- 5\55, 2\95-3\35, - 2\45, - 0\8, Iris-setosa
6\15+, 3\35+, 4\75+, 1\75+, Iris-virginica
6\15+, 2\95-3\35, 4\75+, 1\75+, Iris-virginica
6\15+, - 2\95, 4\75+, 0\8-1\75, Iris-virginica
6\15+, - 2\95, 4\75+, 1\75+, Iris-virginica
5\55-6\15, - 2\95, 4\75+, 1\75+, Iris-virginica
- 5\55, - 2\95, 2\45-4\75, 0\8-1\75, Iris-versicolor
5\55-6\15, 2\95-3\35, 2\45-4\75, 0\8-1\75, Iris-versicolor
6\15+, 2\95-3\35, 4\75+, 0\8-1\75, Iris-versicolor
6\15+, - 2\95, 2\45-4\75, 0\8-1\75, Iris-versicolor
6\15+, - 2\95, 2\45-4\75, 0\8-1\75, Iris-versicolor
- 5\55, 2\95-3\35, - 2\45, - 0\8, Iris-setosa
- 5\55, 3\35+, - 2\45, - 0\8, Iris-setosa
- 5\55, 3\35+, - 2\45, - 0\8, Iris-setosa
5\55-6\15, 3\35+, - 2\45, - 0\8, Iris-setosa
- 5\55, 3\35+, - 2\45, - 0\8, Iris-setosa
6\15+, 2\95-3\35, 4\75+, 1\75+, Iris-virginica
5\55-6\15, - 2\95, 4\75+, 1\75+, Iris-virginica
6\15+, 2\95-3\35, 4\75+, 1\75+, Iris-virginica
6\15+, 3\35+, 4\75+, 1\75+, Iris-virginica
6\15+, 2\95-3\35, 4\75+, 1\75+, Iris-virginica
5\55-6\15, 3\35+, 2\45-4\75, 0\8-1\75, Iris-versicolor
6\15+, 2\95-3\35, 2\45-4\75, 0\8-1\75, Iris-versicolor

Unstructured data Reuters-21578

Part of training data Reuters-21578

@1939

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute Word

@attribute class

@data

affiliate, sell, unit, york, june, corp, signed, definitive, agreement, pty, group, undisclosed, terms, preliminary, reached, march, completion, sale, approval, shareholders, stock, exchange, founded, wholly, owned, manufactures, markets, products, pct, black, company, acq

financial, buys, stake, april, systems, chairman, sold, common, shares, corp, undisclosed, terms, executive, officer, company, pct, control, martin, board, directors, acq

boeing, merger, period, june, required, tender, offer, argosystems, midnight, dlr, share, cash, electronics, firm, acq

general, partners, sells, gencorp, stake, washington, april, partnership, recently, ended, bid, securities, exchange, commission, sold, remaining, pct, company, shares, share, market, transaction, york, stock, sale, common, includes, industries, week, dlr, hostile, tender, offer, acq

acquires, stores, june, acquired, undisclosed, amount, cash, acquisition, number, owned, company, acq

chrysler, pact, april, corp, agreed, period, definitive, agreement, proposed, billion, dlr, takeover, letter, intent, signed, march, date, reached, companies, plan, deal, additional, prior, due, diligence, investigation, company, talks, statement, terminated, official, part, donaldson, lufkin, jenrette, analyst, henderson, acq

buyout, bid, june, products, management, group, withdrawn, dlr, share, leveraged, offer, due, continued, results, terms, financing, led, price, current, acq



dayton, hudson, buyer, stock, interested, acquired, acq
 plans, sell, unit, june, ended, companies, week, reached, agreement, principle,
 purchase, largest, terms, care, hmo, intends, acq
 ccr, offer, takeover, talks, los, angeles, oct, video, corp, received, investment,
 vancouver, acquire, controlling, company, tender, terms, board, support, additional,
 details, acq
 becor, western, talks, bidder, june, company, lynch, corp, offer, withdrawn, week,
 board, evaluate, plans, today, adjourn, stockholders, merger, agreement, buyout,
 acquisitions, mining, manufacturing, latest, proposal, calls, pct, stock, holders, retain,
 held, management, half, terms, financial, acq
 corp, completes, acquisition, june, completed, privately, held, terms, disclosed,
 company, san, systems, software, development, sales, operate, part, products, group,
 acq
 jwt, group, plc, york, june, filed, suit, enjoin, company, tender, offer, unit, executive,
 peters, confidential, information, clients, court, seeks, units, january, thompson,
 subsidiary, officer, acquiring, stock, make, gains, acq
 receives, takeover, april, industries, seeking, acquired, recently, received, purchase,
 company, identify, parties, investment, march, engaged, seek, purchasers, units,
 corp, acq
 standstill, accord, los, angeles, april, group, reached, agreement, resources, parent,
 companies, acquiring, pct, business, combination, approved, board, company,
 advised, owns, outstanding, common, stock, addition, agreed, vote, shares,
 arrangement, tendering, securities, owned, tender, offer, acq
 industries, buys, business, june, purchased, utility, cash, details, transaction,
 disclosed, annual, sales, sold, formed, subsidiary, manufactures, sells, acq
 total, buys, mining, vancouver, june, resources, standard, purchased, dome, shares,
 cash, companies, acq



 Universiti Utara Malaysia

errill, lynch, qtr, shr, cts, blah, earn
 loss, april, quarter, ended, bank, chairman, chief, executive, company, profits,
 earned, cent, share, reported, profit, cts, compared, earlier, president, michael, james,
 subordinated, due, agreement, agreements, outstanding, line, reserve, end, day, field,
 sales, force, representatives, dealers, employees, reduce, fixed, statement, affected,
 move, states, continue, director, added, reuter, earn
 genetics, higher, losses, cambridge, mass, april, earlier, increased, quarter, net, loss,
 reported, ended, compared, company, result, strategic, decision, levels, equity,
 development, products, bring, market, february, rose, reuter, earn
 split, april, directors, stock, common, payable, shareholders, record, reuter, earn
 qtr, jan, net, april, oper, shr, cts, revs, avg, shrs, operating, excludes, gains, share,
 quarter, tax, loss, carryforwards, reuter, earn
 qtr, sept, mass, oct, shr, cts, net, sales, avg, shrs, reuter, earn
 qtr, net, april, shr, avg, shrs, assets, billion, deposits, loans, pct, results, restated,
 pooled, bank, include, purchase, loss, provision, reuter, earn
 qtr, shr, cts, blah, earn

feed, wheat, tenders, trade, european, community, increased, export, intervention,
 south, korea, destination, traders, tender, originally, tonnes, shipment, poland,
 tranches, grain

onic, tenders, wheat, pakistan, french, cereals, intervention, tender, tonnes, soft, food, aid, programme, official, grain, shipped, european, community, shipment, bulk, grain
china, corn, commitments, usda, washington, tonnes, previous, agriculture, department, export, sales, report, week, additional, resulted, destinations, total, delivery, season, grain
brazil, grain, harvest, storage, sao, paulo, april, crop, tonnes, agriculture, ministry, leonardo, brito, brasilia, year, estimated, normal, loss, harvesting, theoretically, distributed, parana, grande, sul, pct, production, regions, crops, maize, grains, poor, storing, sacks, loose, shortage, sheer, transporting, evident, reports, enormous, queues, waiting, granaries, grain
pakistan, private, cotton, rice, exports, islamabad, pakistani, government, allowed, sector, export, trade, cover, years, planning, mahbubul, haq, televised, import, yarn, main, handled, exclusively, state, corporations, high, quality, local, ancillary, compete, effectively, world, overcome, domestic, shortages, grain
ccc, credit, guarantees, rice, algeria, usda, washington, april, commodity, corporation, authorized, sales, year, export, guarantee, program, agriculture, department, additional, increase, agricultural, eligible, coverage, line, exported, september, aid, grain
export, inspections, thous, bushels, soybeans, wheat, corn, blah, grain
senate, panel, votes, county, loan, rate, crops, blah, grain

Part of testing data Reuters-21578

@relation testRReuters
@770
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word
@attribute Word



@attribute Word

@attribute class

@data

acquires, telephone, june, corp, completed, acquisition, company, terms, disclosed, acq

purchase, west, beach, fla, oct, sold, facilities, corp, developer, business, includes, plan, country, acq

times, buy, news, york, june, agreement, terms, disclosed, company, county, northeast, country, purchase, includes, acq

management, april, international, plan, sell, business, group, satisfactory, financing, company, intends, pursue, corporate, acquisition, alternatives, acq

usair, buy, pct, piedmont, shares, tendered, acq

barnett, banks, files, board, fla, june, bank, filed, suit, savings, insurance, corp, companies, district, court, enjoin, adopted, year, seeks, give, proposed, acquisition, acq

brands, acquisitions, york, april, acquisition, distillers, chemical, corp, business, tobacco, company, merrill, lynch, year, billion, bid, unilever, agreed, buy, candidate, thompson, securities, make, part, morris, reynolds, cash, low, growth, capital, funds, spirits, makes, gilbey, decline, acq

group, cuts, distillers, stake, washington, june, investor, led, family, worth, chemical, corp, shares, pct, total, common, filing, securities, exchange, commission, sold, prices, ranging, share, required, disclose, stock, acq

buys, stake, dallas, june, corp, holds, pct, goods, company, stock, acquired, market, disclosure, acq

life, stake, sold, toronto, june, development, corp, agreed, sell, pct, placement, quebec, fund, company, buy, common, shares, share, transaction, purchasers, plan, acquire, remaining, acq

computer, buy, products, firm, june, services, acquired, cash, industrial, assets, company, acq

investors, computerland, edelman, york, june, investor, group, agreed, buy, computer, retailer, sell, international, corp, today, held, largest, retailing, chain, country, bought, led, warburg, pct, owned, founder, money, management, venture, capital, firm, disclose, transaction, estimated, stores, generated, billion, sales, year, parent, company, officials, reached, comment, funds, retailers, strong, make, forces, service, support, recently, give, control, faber, chairman, executive, officer, plan, acq gaf, corp, management, group, acquisition, proposal, acq

waltham, bank, initial, dividend, mass, april, qtrly, div, cts, payable, record, reuter, earn

plan, white, april, board, adopted, dividend, stock, purchase, common, share, outstanding, company, designed, protect, shareholders, control, making, offer, shares, response, specific, takeover, attempt, buy, initial, exercise, price, rights, approximately, equal, prior, group, acquires, pct, tender, result, entitled, cts, position, acquired, existing, shareholder, buys, additional, transactions, effective, expire, years, details, letter, reuter, earn

qtr, loss, calif, oct, shr, primary, cts, profit, diluted, net, avg, shrs, mths, loans, deposits, assets, prior, mth, include, operating, carryforward, gains, share, reuter, earn

qtr, net, oct, shr, cts, sales, mths, reuter, earn

merrill, qtr, april, net, paul, june, shr, cts, revs, avg, shrs, reuter, earn

loss, oct, shr, cts, profit, net, revs, reuter, earn
atlantic, american, qtr, net, atlanta, oct, shr, profit, cts, loss, revs, mths, includes,
gain, share, gains, cent, charge, reserve, reuter, earn
qtr, sept, net, calif, oct, shr, cts, revs, avg, shrs, reuter, earn
country, jewelry, qtr, net, york, june, shr, cts, revs, quarter, ended, company, full,
reuter, earn
mths, loss, june, ended, shr, nil, profit, net, revs, full, resources, reuter, earn
qtr, net, april, shr, cts, assets, billion, deposits, loans, results, restated, reflect,
acquisition, united, banks, reuter, earn
standard, commercial, qtr, net, june, ended, shr, cts, revs, full, latest, includes, tax,
loss, carryforwards, discontinued, reuter, earn
federal, qtr, oper, net, oct, shr, cts, mths, assets, billion, loans, deposits, operating,
excludes, tax, credits, share, quarter, early, retirement, association, full, company,
reuter, earn
england, bank, qtr, net, london, conn, april, shr, cts, stock, aug, reuter, earn
industrial, payout, april, qtr, div, cts, prior, pay, july, record, june, reuter, earn

lawson, interest, rate, prospects, unchanged, london, oct, chancellor, exchequer,
nigel, collapse, share, week, implication, domestic, rates, television, interview, past,
days, upward, pressure, sterling, stayed, crisis, strong, economic, pct, bank, base,
lending, analysts, frantic, financial, shares, reuter, interest
sumita, discount, rate, cut, central, bank, blah, interest
sallie, mae, adjusts, discount, rates, notes, maturity, rate, days, pct, reuter, interest
japan, ease, credit, bank, policy, told, reuters, responding, bond, market, central, cut,
pct, discount, rate, prime, leaves, governor, satoshi, sumita, osaka, early, week,
impossibility, holiday, reuter, interest
concerned, interest, rate, rise, greenspan, blah, interest
marine, midland, bank, cuts, prime, rate, pct, effective, immediately, blah, interest
commonwealth, bank, cuts, australian, prime, sydney, australia, lower, rate, pct,
overdraft, effective, trends, key, lending, longer, term, latest, cut, rates, recent, days,
decline, market, range, reuter, interest
money, market, stg, london, bank, england, morning, system, central, outright, bills,
band, pct, reuter, interest
fdic, seidman, higher, rates, banks, oct, federal, deposit, corp, concerned, impact,
sharp, rise, interest, attending, bankers, convention, expect, economy, banking,
greater, rate, rises, concern, told, news, conference, reuter, interest
central, bank, yields, rise, certificates, deposit, higher, monday, offering, rose, point,
pct, maturities, reuter, interest
bank, france, leaves, intervention, rate, unchanged, pct, official, blah, interest
analysts, doubt, fed, firmed, borrowing, rise, cherrin, reuters, york, economists,
federal, reserve, firming, policy, aid, dollar, higher, discount, window, borrowings,
latest, period, wednesday, today, show, net, averaged, funds, high, pct, case, support,
averaging, economist, noted, pushes, borrowings, argue, catchup, mccarthy, capital,
markets, spokesman, told, conference, week, caused, add, fewer, reserves, needed,
market, days, added, temporary, indirectly, monday, customer, repurchase,
agreements, supplied, system, repurchases, tuesday, put, overnight, repos, clear,
time, leuzzi, afford, lift, interest, rates, weak, economies, abroad, financial, stress,
countries, tightened, tumbled, precipitous, drop, yesterday, monetary, substantive,

fact, currency, dealers, prevailing, yen, huge, japan, aiming, steady, average, rate, early, suspect, reuter, interest
interest, higher, rates, disbursed, credit, month, pct, rate, treasury, noted, reuter, interest

APPENDIX B

Google stop word Removal List

“a about above after again against all am an and any are are not as at be because been before being below between both but by ca not cannot could could not did did not do does does not doing do not down during each few for from further had had not has has not have have not having he he'd he'll he's her here here's hers herself him himself his how how's i i'd i'll i'm i've if in into is is not it it's its itself let's me more most must not my myself no nor not of off on once only or other ought our ours ourselves out over own same sha not she she'd she'll she's should should not so some such than that that's the their theirs them themselves then there there's these they they'd they'll they're they've this those through to too under until up very was was not we we'd we'll we're we've were were not what what's when when's where where's which while who who's whom why why's with wo not would would not you you'd you'll you're you've your yours yourself yourselves”

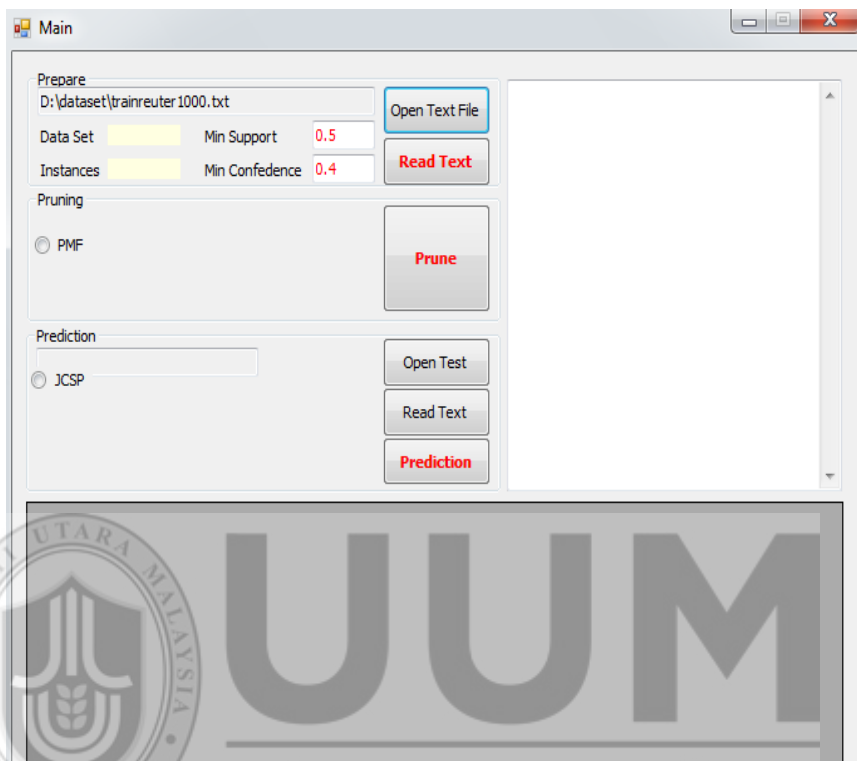
APPENDIX C

Rule generate

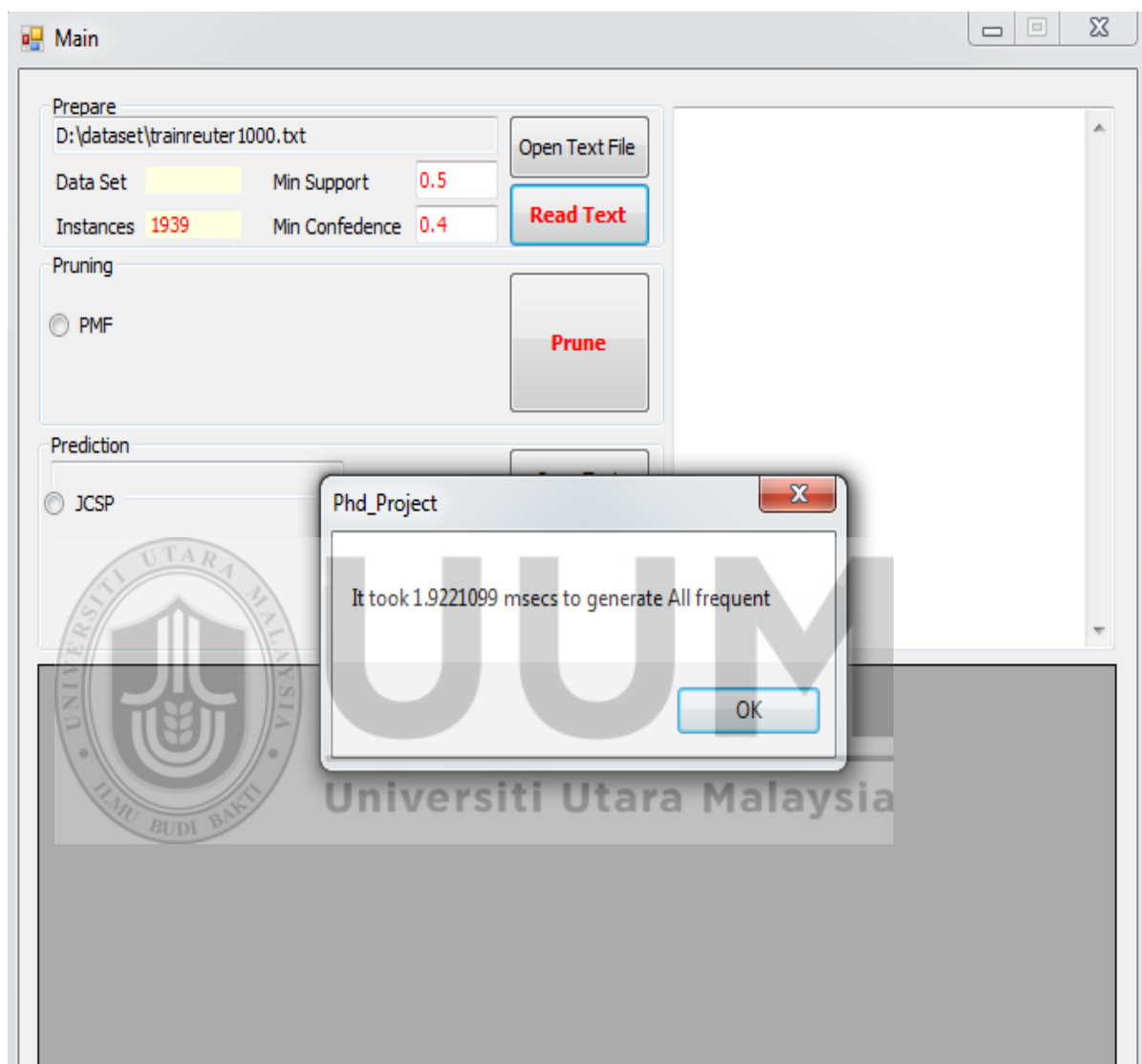
ID	NoOfTerm	Class_ID	Term_ID	TermDesc	TermCount	Rule_Desc	ClassDesc	ClassCount	RuleSub	RuleConf	IsCandidate
22732	1	2078	37956	shares	999	shares==>acq	acq	1377	9.7923818854661	1	0
22733	1	2078	37957	company	833	company==>acq	acq	1377	0.1361278649664	1	0
22734	1	2080	37958	net	1812	net==>earn	earn	2717	0.2842896844858	0.99971302428254	0
22735	1	2078	37959	cs	2073	cs==>earn	earn	2717	0.311862012386794	0.9792571152918	0
22736	1	2078	37960	year	206	year==>acq	acq	1377	3.36766388752651	1	0
22737	1	2078	37961	offer	337	offer==>acq	acq	1377	5.50923653386627	1	0
22738	1	2078	37962	stake	329	stake==>acq	acq	1377	5.37845349027301	1	0
22739	1	2078	37963	merger	244	merger==>acq	acq	1377	3.98888343959451	1	0
22740	1	2080	37964	loss	1005	loss==>earn	earn	2717	0.15661271865291	0.9532338308457	0
22741	1	2078	37965	acquisition	387	acquisition==>acq	acq	1377	6.32663070132431	1	0
22742	1	2079	37966	crude	180	crude==>crude	crude	379	2.9426189308484	1	0
22743	1	2079	37967	prices	153	prices==>crude	crude	379	2.50122609122111	1	0
22744	1	2079	37968	oil	350	oil==>crude	crude	379	5.72175903220531	1	0
22745	1	2079	37969	barrels	140	barrels==>crude	crude	379	2.28870361288211	1	0
22746	1	2080	37970	shr	1480	shr==>earn	earn	2717	0.24194866764791	1	0
22747	1	2080	37971	qt	1230	qt==>earn	earn	2717	0.20107896027461	1	0
22748	1	2080	37972	pct	697	pct==>earn	earn	2717	6.71817901218091	0.5896700143472	0
22749	1	2080	37973	profit	767	profit==>earn	earn	2717	0.12531828222001	1	0
22750	1	2080	37974	revs	957	revs==>earn	earn	2717	0.15944823822441	1	0
22751	1	2084	37975	trade	366	trade==>trade	trade	367	0.05656387510492	0.9433551912568	0
22752	1	2083	37976	bank	679	bank==>Money-FX	Money-FX	503	5.39480137322211	0.466008365243	0
22753	1	2081	37977	agriculture	195	agriculture==>gran	gran	429	1.8178371758831	1	0
22754	1	2081	37978	wheat	226	wheat==>gran	gran	429	3.61462154659371	1	0
22755	1	2081	37979	corn	140	corn==>gran	gran	429	2.28870361288211	1	0
22756	1	2081	37980	gran	185	gran==>gran	gran	429	3.02430074559421	1	0
22757	1	2082	37981	lennet	324	lennet==>gran	gran	429	3.6625238618141	1	0
22758	1	2082	37982	rates	146	rates==>interest	interest	345	2.28679191057701	1	0
22759	1	2083	37983	money	328	money==>Money-Money-FX	Money-FX	503	3.08974887739081	0.57621891218951	0
22760	1	2083	37984	market	483	market==>Money-Money-FX	Money-FX	503	0.04953488533591	0.62732918254651	0
22761	1	2082	37985	rate	225	rate==>interest	interest	345	3.67827366356051	1	0
22762	1	2083	37986	yen	134	yen==>Money-FX	Money-FX	503	2.18061631518731	1	0
22763	1	2083	37987	currency	180	currency==>Money-Money-FX	Money-FX	503	2.9426189308484	1	0
22764	1	2083	37988	exchange	193	exchange==>Mon-Money-FX	Money-FX	503	3.15514140918751	1	0
22765	1	2083	37989	central	176	central==>Money-Money-FX	Money-FX	503	2.87722739905181	1	0
22766	1	2083	37990	dollar	234	dollar==>Money-FX	Money-FX	503	3.82540461010291	1	0
22767	1	2084	37991	dealers	134	dealers==>Money-Money-FX	Money-FX	503	2.18061631518731	1	0
22768	1	2084	37992	bilion	177	bilion==>trade	trade	367	2.89357528200091	1	0
22769	1	2084	37993	exports	137	exports==>trade	trade	367	2.23965996403461	1	0
22770	1	2084	37994	surplus	124	surplus==>trade	trade	367	0.02027137485691	1	0
22771	1	2084	37995	japan	145	japan==>trade	trade	367	2.33044302762791	1	0
22772	1	2084	37996	deficit	128	deficit==>trade	trade	367	2.01252961748221	1	0
22773	1	2084	37997	countries	131	countries==>trade	trade	367	2.14157266633971	1	0

APPENDIX D

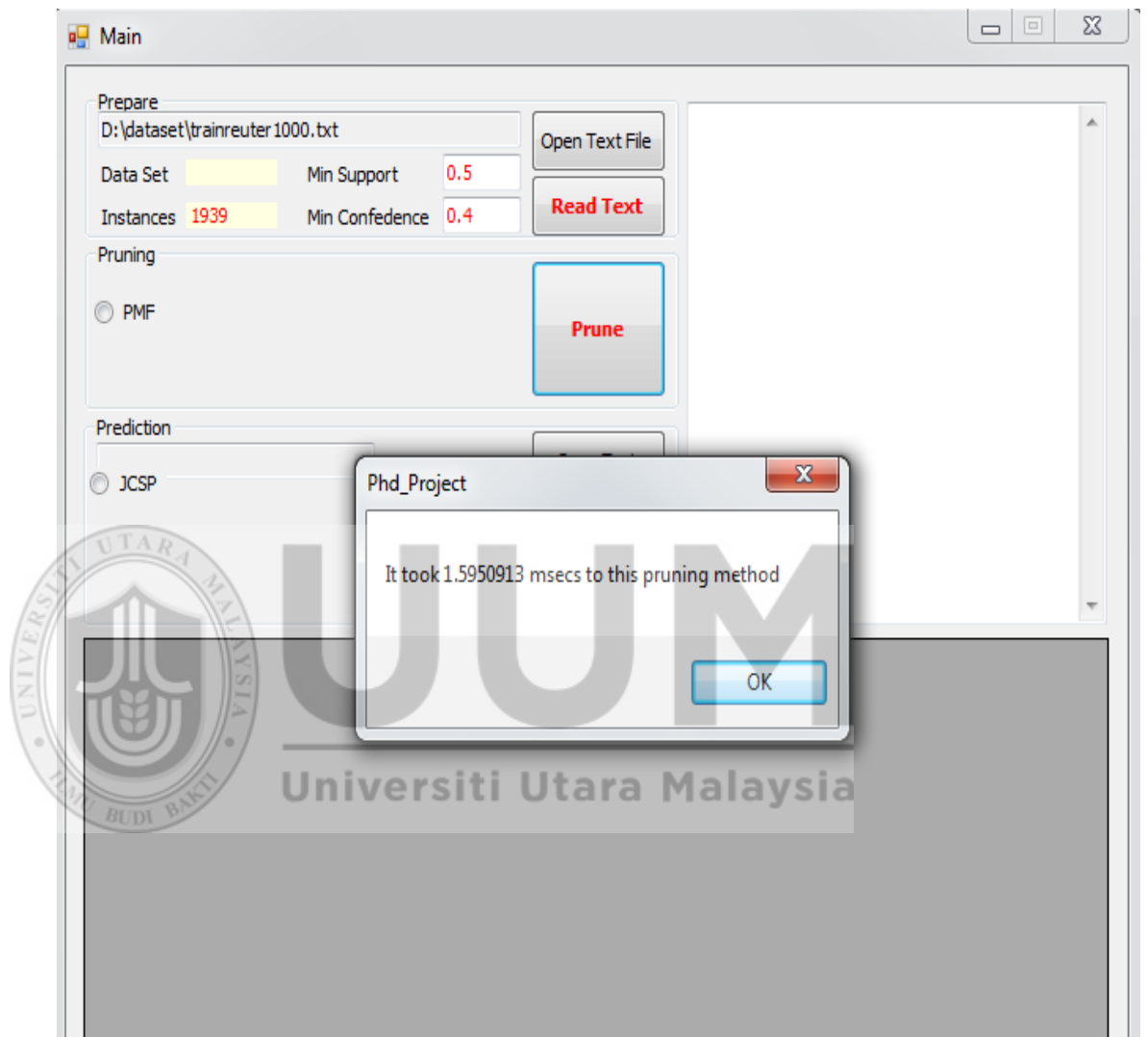
Screen shot of Classifier



Screen Shot Time to Generate Rule



Screen shot Pruning



Screen shot Prediction

