Reducing Rule Generation Complexity in the Prism Algorithm

By
Mohammed Yaseen Hammadi

Supervisor
Dr. Rashid Al-Zubaidy

This Thesis was Submitted in Partial Fulfilment of the Requirements for the Master's Degree in Computer Science

Deanship of Academic Research and Graduate Studies
Philadelphia University

May 2014

جامعة فيلادلفيا
نموذج تفويض

أنا محمد ياسين حمادي، أفوض جامعة فيلادلفيا بتزويد نسخ من رسالتي للمكتبات أو المؤسسات أو الهيئات أو الأشخاص عند طلبها.

التوقيع:
التاريخ:

Philadelphia University
Authorization Form

I am, Mohammed Yaseen Hammadi, authorize Philadelphia University to supply copies of my thesis to libraries or establishments or individuals upon request.

Signature:

Date:

Reducing Rule Generation Complexity in the Prism Algorithm

By
Mohammed Yaseen Hammadi

Supervisor
Dr. Rashid Al-Zubaidy

This Thesis was Submitted in Partial Fulfilment of the Requirements for the Master's Degree in Computer Science

Deanship of Academic Research and Graduate Studies
Philadelphia University

May 2014

Successfully defended and approved on _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Examination Committee Signature		**Signature**
Dr. Academic Rank:	, Chairman.	_ _ _ _ _ _ _ _ _ _ _ _
Dr. Academic Rank:	, Member.	_ _ _ _ _ _ _ _ _ _ _ _
Dr. Academic Rank:	, Member.	_ _ _ _ _ _ _ _ _ _ _ _
Dr. Academic Rank:	, External Member.	_ _ _ _ _ _ _ _ _ _ _ _

Dedication

First of all I thank Allah Almighty for giving me the strength and knowledge to finish this work. I dedicate this work to my family: wife and kids, also to my friends, they've been there for me whenever I needed.

Mohammed Yaseen Hammadi
2014

Acknowledgment

(قَالُواْ سُبْحَانَكَ لاَ عِلْمَ لَنَا إِلاَّ مَا عَلَّمْتَنَا إِنَّكَ أَنتَ الْعَلِيمُ الْحَكِيمُ)

It would not have been possible to write this master thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

Above all, I would like to express my thanks and sincere gratitude for who has guided me through my study and my thesis work; my supervisor *Dr. Rasheed Al-Zubaidy*, also I was honoured to benefit from the comments and guidance *of Prof. Saeed Al-Ghoul*, *Dr. Nameer El-Emam, and Dr.Ali Alawneh, and Dr. Rehab Duwairi*.

Finally, I would like to thank my parents, wife, kids, friends for giving me their unequivocal support, and to those who supported and encouraged me in any way; my teachers, superiors and colleagues at Philadelphia University.

Mohammed Yaseen Hammadi

2014

List of Figures

List of Tables

Abstract

Data mining is a computer science field that works on finding the relations and patterns that are found within data (training data), by detecting these relations and patterns, rules can be created, that can be used later on to filter out and process future data (test data). Prism is an easy covering algorithm depending on separate-and-conquer algorithms; this algorithm creates rules by discovering the power of relations between attribute items with the objective class. The primary phase in Prism is the rule creation phase, where the relation power between every attribute item and the targeted class is computed in every assumption, then the set of training data is classified based on the output results.

One of the primary pillars in the rule generation step is that whenever Prism detects two equal strength values, Prism selects one value only and drops the other to the next iteration of computations and filtering. However, observations and experiments over more than one data set proved that in each time the other equal value is always chosen in the next iteration, this is obviously unneeded system overhead.

In this thesis, we aim to remove this redundancy in rule generation phase by presenting the enhanced prism (E-Prism) algorithm.

Another insufficiency in this algorithm is that it deals with only categorical attributes, some discretization methods where previously used with Prism, but the problem with these techniques is their large calculation complexity compared to other discretization methods. This thesis aims overcome this problem by using discretization methods with small complexity as a pre-processing phase allowing Prism to deal with continuous attributes.

Chapter One: Introduction

1.1 Introduction

In this chapter, we show an overview of the main issues that were handled in this thesis, the importance of the thesis, the main structure and its contribution.

Data mining is an approach to analyze data and to transform this raw data into helpful information. This approach is applied in many different areas such as marketing applications, business intelligence applications that help users to make better business decisions, in Internet search technology and multimedia. Data mining is a subject domain that joins knowledge and uses the science of statistics, artificial intelligence and natural language processing.

1.2 Background (Prism Algorithm)

In the field of data mining Prism is a rule induction algorithm, which was introduced by (Cendrowska, 1987), and was enhanced by (Bramer, 2002) and (Stahl and et.al, 2009), the main goal of this algorithm is to make an immediate sorting rules from training sets. Each rule is established by the algorithm by creating it term-by-term and selecting the attribute-value pair that increases the probability of a selection outcome class. Each term is defined as "attribute = value" form.

In the original format, the Prism algorithm utilizes separate-and-conquer technique, there are specific steps starting with determining the probability of each classification for all pairs, then selecting the highest probability pair and initiate a subset to add it to rule set to create the rule, after that repeat the previous steps until finally having a subset that contains the rule for classification.

Prism algorithms follow the separate and conquer approach; it addresses some of the shortcomings of decision trees, such as the replicated sub tree problem. Its main steps are as follow:

- As a starter, a rule is declared.
- After that, all items related to the rule are sorted out.
- Then remaining instances are conquered.

After this phase, rule established classification become a common model in data mining where the result is outlined in "If-Then" format and saved in the learning principle, while the challenge of classical rule based classification particularly in induction algorithms like RIPPER is specific size classifiers (specific count of attributes creating a rule) with often low accuracy. In the other side, Prism algorithm creates high numbers of rules and often attempts to get optimal rules. Prism also does not handle Continuous data.

1.3 Discretization of Continuous Attributes

It is the technique of dividing the continuous attributes into discreet attributes (E Xu, et al, 2010). Unfortunately, the total number of methods to discretize the continuous attribute is unlimited. Data discretization is a general, goal driven pre-processing technique that is used to create groups of value ranges for a specific continuous variable by partitioning its domain into a predefined number of disjoint ranges, and then relates these ranges with reliable labels (Sheng-yi J. et.al, 2009). Subsequently, data are resolved at this larger stage of knowledge representation rather than the exact individual values, in order to achieve a very easy representation of the data in data exploration and data mining technique. A discretization technique flows in mainly four steps as defined in (Figure 1). (Joao G, et al, 2006).

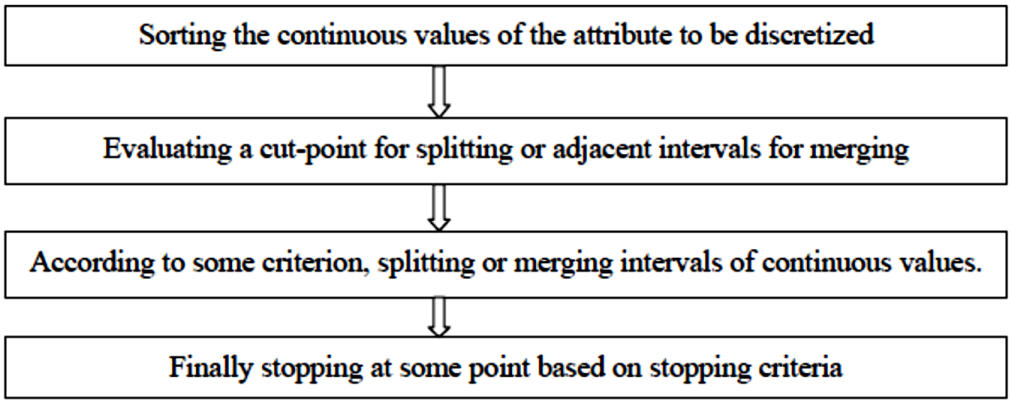

Figure 1 Steps of discretization

The main purpose of discretization is to discover a group of cut points to divide the domain into a little number of ranges. Mainly there are two ways of discretization. The first is to calculate the number of discrete ranges, or the user should mention the number

of ranges, the other, is to calculate the edges of the specific ranges, the range of values of a continuous attribute. Usually, in this technique, after classifying the goal class data (in ascending or descending) with respect to the goal class, data break points are assigned through the dataset items. In general, the algorithm for selected break points can either begin with empty groups which means a top down approach using split divides, or the other way bottom-up, which begins with the list of all the values as break points and combines the ranges. In both cases, there is a stopping criterion, which explains when to stop the discretization manner (Mitov L et al, 2009).

The main advantages of data discretization can be defined in different ways:

- The experts always explain variables using language terms instead of a real value. In other words, the discretization supplies better perceiving of attributes.
- The total number of data can be largely minimized because some redundant data can be deleted.
- It supplies enhanced performance for the rule extraction.

1.4 Discretization methods

The motive for the discretization of continuous attributes is based on the necessity to achieve larger accuracy ranges in order to maintain data with large cardinality attributes. Discretization methods have been enhanced through various views due to various necessities, Table 1 shows a comparison between some discretization techniques (Rajashree D et al, 2011).

Table 1 : Comparing main descritization techniques

Methods / **Evaluation**	**Equal Width**	**Equal Frequency**	**K-means Clustering**	**Entropy Based**	**Chi Merge based**
Supervised/ Unsupervised	Unsupervised	Unsupervised	Unsupervised	Supervised	Supervised
Dynamic/ Static	Static	Static	Static	Static	Static
Global/ Local	Global	Global	Local	Local	Global
Splitting/ Merging	Split	Split	Split	Split	Merge
Direct / Incremental	Direct	Direct	Direct	Incremental	Incremental

Stopping Criteria	Fixed Bin no.	Fixed Bin no.	No assign data values to given cluster no.	Threshold / Fixed no. of intervals	Threshold / Fixed no.
Sensitive to outlier	Yes	No	Yes	No	No
Same values to different intervals	No	Yes	No	No	No
for Complexity attribute of n objects	O(n)	O(n)	O(ikn) i= iteration k= intervals	O(n log(n))	O(n log(n))

The methods of Discretization can be classified to:

1- Supervised and Unsupervised methods: they utilize the class label when dividing the continuous attributes. It may be described as error-based, entropy-based or statistics-based depending on whether the ranges that are chosen by metrics based on error or on the training data set, entropy of the ranges, or some statistical calculations.

However, unsupervised discretization methods do not request the class information to divide continuous attributes; it partitions the continuous attributes into sub-intervals. It is implemented in very early techniques such as equal-width and equal-frequency (Dougherty et al, 1995).

These manners may not achieve best outputs in cases where the distribution of the continuous parameters values are not regular. If no class information is obtainable, unsupervised discretization is the only option. In supervised discretization manners, class information is utilized to discover the suitable ranges raised by cut-points. Various ways have been defined to utilize this class information for discovering helpful ranges in continuous attributes.

2- Dynamic or Static methods: A dynamic method will discretize the continuous attributes when a sorting is being created, such as in C4.5. It is mutually connected with corresponding classification method, where algorithm can work with real attributes, but in the static methods discretization is made in preprocessing phase to the rating (Quinlan, J.R et.al,1993).

3- Global and Local methods (Dougherty et al, 1995): they are related to the stage when the discretization occurs. Global approaches discretize parameters before the rule generation starts, it utilizes the whole data to discretize. On the other hand, local methods discretize attributes during the induction process. Experimental results have defined that global discretization approaches usually produce better results comparing to the local approaches.

1.4.1 Unsupervised Discretization Methods

Through the unsupervised discretization approaches, they are the easiest ones (equal-width and equal-frequency domain binning) and the much advanced ones, established on the clustering anatomy like k-means discretization. Continuous intervals are partitioned into sub intervals by the user fixed width or frequency (Daniela J, 2010).

1.4.1.1 Equal Width Interval Discretization

Equal-width range discretization is the easiest discretization approach that splits the continuous data into k with the same bins size, where k is a feature supplied by the user. The process involves classifying the values of a continuous parameters and discovering the minimum V min and maximum V max values. The range can be calculated by splitting the range of spotted values for the variable into k equally sized bins.

1.4.1.2 Equal-Frequency Interval Discretization

The equal-frequency algorithm calculates the lowest and highest values of the discretized attributes, classifies all values in a specific order; ascending order, and splits the sorted continuous values into k ranges such that every range consists of around n/k data cases with adjacent values. For equal-frequency, much situations of a continuous value might be appointed into many various bins. This algorithm wants to minimize the problems of the equal-width interval discretization by partitioning the domain into ranges with the same distribution of data points. The data cases with corresponding value should take place in the similar domain, thus it is not always possible to create immediately k equal frequency domains. This method is also named as proportional k-interval discretization.

A continuous attribute can split into domains with similar width (figure 2) or similar frequency (figure 3). Other techniques defined to shape the domains, for example concerning on the clustering principles such as K-means clustering discretization.

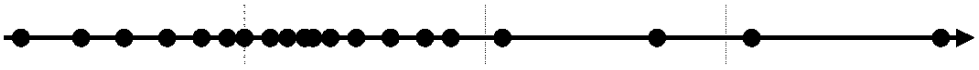

Figure 2 Equal Width Discretization

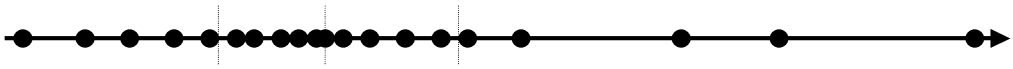

Figure 3 Equal Frequency Discretization

1.4.1.3 Clustering Based Discretization

The k-means clustering approach remains one of the very famous clustering techniques, it is also appropriate to be utilized to discretize continuous valued parameters because it computes continuous distance-based similarity calculations to cluster data points (Sellappan, Tan K. H, 2009).

Moreover, since unsupervised discretization includes just a single variable, it is parallel to a "1-dimensional" k-means clustering appointed. K-means is a non-hierarchical dividing clustering algorithm that makes specific group of data points and supposes that the amount of clusters to be calculated (k) is given.

First, the algorithm specifies random k data points to be the so named centers of the clusters. Then every data point of the specific group is associated to the nearest center resulting the primary distribution of the clusters. Next two steps of this step are explained until the convergence occurred:

1. Calculate the middle of the clusters again as the average for all values in every cluster.

2. Every data point is appointed to the nearest centroid. The clusters are created again.

1.4.2 Supervised Discretization Methods

Supervised discretization approaches make use of the class label when dividing the continuous attributes. Through the supervised discretization approaches, there are the easy ones like Entropy-based discretization approaches and Interval Merging and Splitting using Chi-square ($\chi 2$) Analysis.

1.4.2.1 Entropy Based Discretization Method

It is defined by (Fayyad, Irani, 1993). An entropy-based approach will utilize the class information entropy of candidate splits to choose edges for discretization. Class information entropy is a calculation of pure class sets; it computes the number of values which would be necessary to specify which class the case belongs to. It considers one big domain consisting all defined values of parameters and then again splits this domain into little sub-domains until some stopping criterion is met.

1.4.2.2 Chi-Square Based Discretization

Chi-square ($\chi 2$) behaves as an importance examiner on the relationship through the values of an attributes. The $\chi 2$ statistic calculates the same attribute of adjacent domains based on some significance stage. It tests the hypothesis that two adjacent stages of an attribute are independent of the class. If they are independent, they should be combined; otherwise they should remain disjoint. The $\chi 2$ stopping rule is depending on a user-defined $\chi 2$ threshold to reject the partitioning if the two sub-domains are the same.

The Chi-Merge algorithm (Kerber, 1992) is initialized by sorting the training data depending on their value for the attribute being discretized and then structuring the initial discretization, where every case is put into its own stage. Initially, if two adjacent intervals have a very similar distribution of classes, these stages can be combined. In Chi-Merge, every distinct value of a numerical attribute is considered to be one stage. Then $\chi 2$ tests are defined for every pairs of adjacent intervals and adjacent intervals with least $\chi 2$ values are combined together. This combining process proceeds repeatedly, in order to have a stopping criterion met i.e. until having two values of all adjacent pairs exceeds a

threshold or a specific number of stages has achieved. The threshold is calculated by the significance stage and degrees of freedom = (number of classes -1).

The main problem of Chi-Merge algorithm is that it cannot be utilized to discretize data for unsupervised learning jobs. In addition, it is only trying to define first order correlations, thus could not perform in a right way when there is a second-order correlation without a corresponding first-order correlation, which might happen if an attribute only correlates in the presence of some other condition.

1.5 Motivation

The traditional Prism algorithm suffers from redundancy in its rule generation phase, this is considered wasted time regarding the fact that this redundancy can be avoided.

Another problem to be searched, is the problem of handling continuous data, the traditional Prism cannot handle this type of data, some approaches were implemented to solve this, in this thesis we will try to use different approaches (Equal-frequency and Entropy) to get better results.

The traditional Prism algorithm is suffering from redundancy in one of its main phases (rule generation phase) this is very crucial when we are handling big data, in data mining the typical case is having big data, so any enhancement in this area will be valuable to reduce time and space complexity.

Another major motivation is that the traditional Prism is not able to handle continuous data. Some researchers proposed pre-processing approaches for Prism to handle continuous data, the problem with these methods is that they did not consider time as a main priority in their chosen discretization methods.

1.6 Problem Statement

The main phase in Prism is rule generation, where the relation strength between each attribute item and the targeted class is calculated in each iteration, then the training data set is filtered depending on the results. One main concern in the rule generation step is that whenever Prism detects two equal strength values, Prism chooses one only and leaves the other to enter a new iteration of calculations and filtering. However, tests show

that every time this other value is selected in the next iteration, so why calculating and filtering again , this is obviously an unnecessary overhead, time is a very critical issue in data mining considering the fact that data mining works with big data, so removing this redundancy is very critical to Prism time efficiency.

The traditional Prism uses attribute items to detect the relation power with the target class, this is possible with discreet data but doesn't work with continuous data so a pre-processing discretization step is needed to create discrete intervals. Some discretization techniques were previously used with Prism, the problem with these techniques is their high calculation complexity compared to other discretization techniques, up to our knowledge, no previous work was done in the area of testing Prism with low complexity discretization techniques, in this research we will test that.

1.7 Thesis Objectives

In this research we are enhancing the rule generation phase in the Prism algorithm, traditional Prism works on finding the highest probability value for each item, then adds this selected item to the rule, the problem is that when probability values are equal, Prism chooses one of them randomly or based on a predetermined value, the un-chosen value will be obviously selected in the next filtering round, because as we said before it has the same probability value as the previously selected item, so to solve this redundancy, we will rebuild the rule generation choosing step to make it process both items in the same iteration, saving the time needed to recalculate all other items again.

This research aims to meet the following objectives:

- Enhance the phase of rule generation in order to reduce processing time, which will be utilized in all prism family algorithms.
- Try to handle continuous data with low complexity methods by using proper discretization techniques with prism algorithm in order to handle continuous attributes.

1.8 Thesis Contributions

Our contribution in this thesis is derived from an extensive study on the insufficiencies in the Prism algorithm, we summarized our contributions as follows:

- Our work aims at reducing the time complexity for the Prism algorithm, mainly enhancing the rule generation step in it.
- Another contribution will be testing Prism with less complex pre-processing discretization methods, as the previously used methods are complex.

1.9 Thesis Methodology:

In order to solve the problems mentioned above and achieve our contributions, we will follow the following methodology:

First, we have to study Prism algorithm, then we will study discretization methods to recognize weaknesses and strengths in each method, after that we will select the proper method to implement it with the Prism algorithm, after this step prism can deal with continuous attributes. Then we will fix the following problem in the rule generation phase.

- When prism algorithm starts to generate rules and begins to filter the dataset to get probability values for items, prism will then select the highest probability from the items and places this item in the rule.

Assume that A=v1 and B=v2 are items. Assume further that the probability of A=v1 and B=v2 are equal. In traditional Prism, either A=v1 or B=v2 will be added to the premises of the rule in iteration i. in the next iteration, experimental evidence has shown that the left out item will be chosen in the next iteration (i.e. iteration i+1).

In the traditional prism algorithm we will select randomly (arbitrarily) one of these items and add it to the rule, and in the next step of filtering, Prism recalculates again and it will obviously select the second item that appeared in the previous step of filtering because it was the highest and will add it to the rule, illustrated in figure (4), this method causes wasted time to the algorithm.

The rule in the first step will be (IF A=v1 then D) where A= Attribute.

The rule in the second step will be (IF A=v1 AND B=v2 then D).Where B= Attribute, D= class.

To solve this problem we can merge two steps in one step, as illustrated in figure (5).

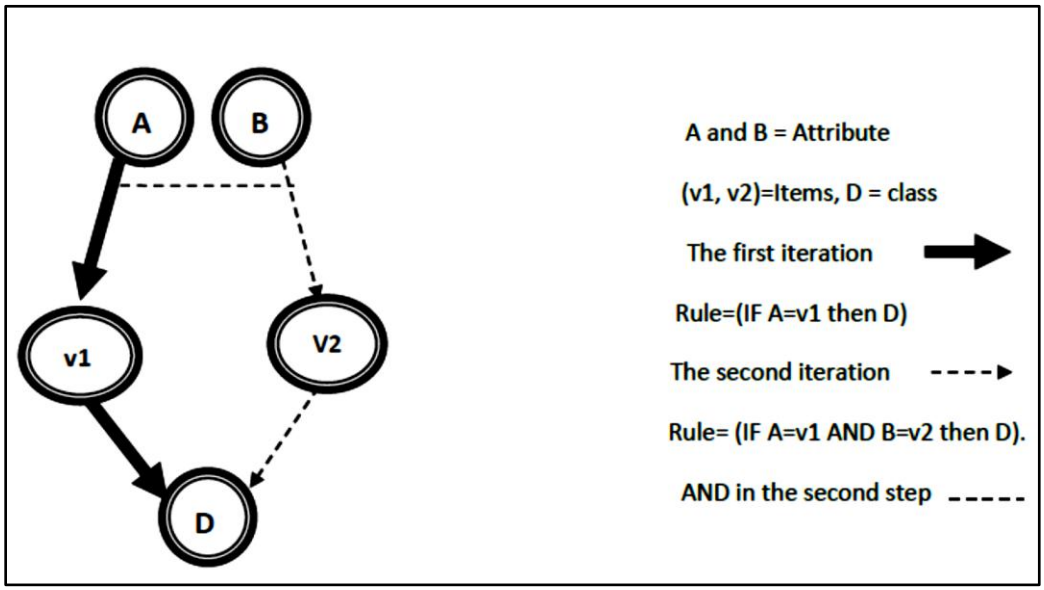

Figure 4 Generate rule in traditional Prism

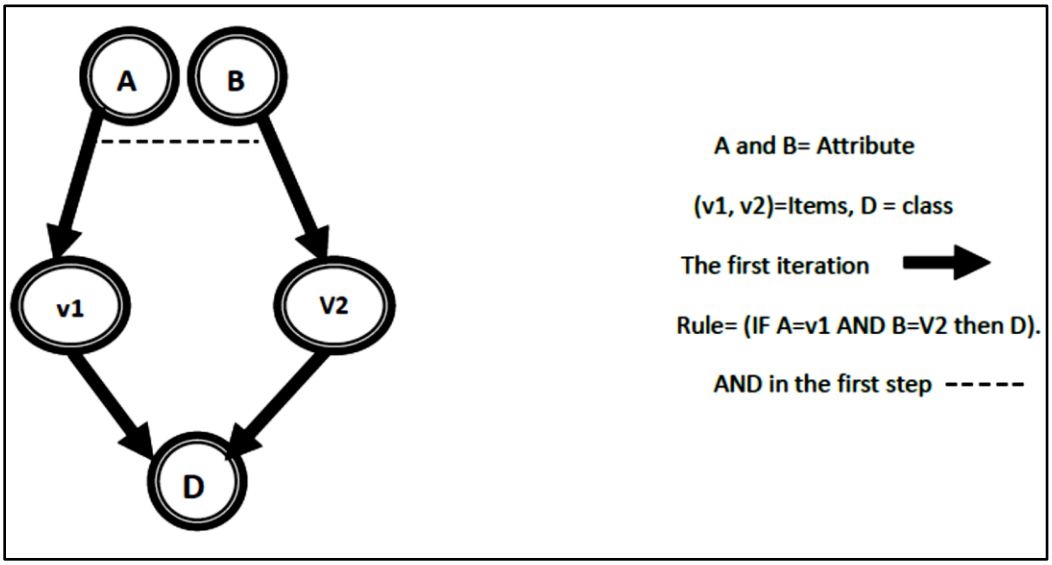

Figure 5 the proposed enhancement generate rule

After this step, we will try to enhance rule generation and if we find any weakness in it, then we will try to overcome the weakness, then we will generate the rule, as illustrated in (figure 6)

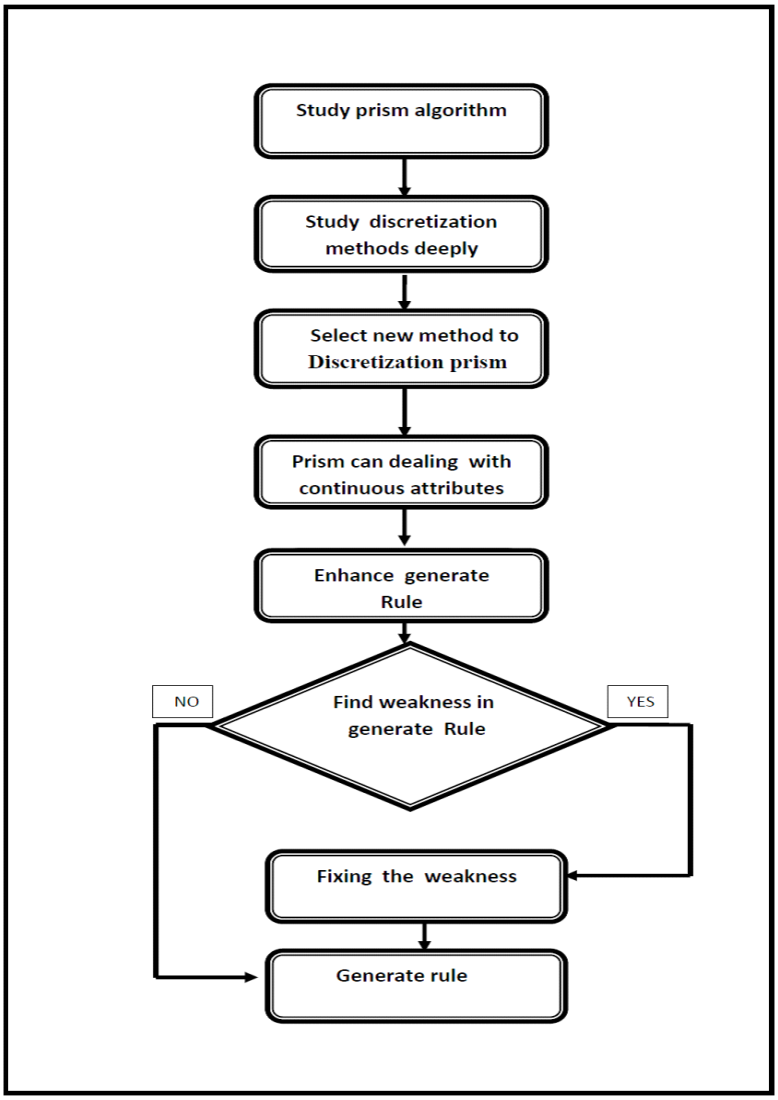

Figure 6 Block diagram for thesis methodology

1.10 Thesis Outline

This thesis consists of five chapters. Chapter two displays the literature review and related works of our enhanced technique, we will talk about the classification of data mining that consists of simple one rule, Divide-and Conquer approaches that consist also of other classification, Statistical Approach (Naïve Bayes), Separate-and Conquer approaches and its classification and finally about Hybrid Approach.

In Chapter three we will talk about the problem statement that shows that the Prism algorithm suffer from redundancy in its rule generation phase, this is considered wasted

time regarding the fact that this redundancy can be avoided. Another problem, the algorithm can't handle continuous attributes, and it has another insufficiency in generating rules, this algorithm generates rules step by step which increase time to generate rules.

In Chapter four we will display the implementation requirement and results. We can implement our work on various domain data that have been previously tested with other similar algorithms, in that way we will be able to detect the enhancement in our algorithm.

Finally, in Chapter five we will show the conclusion of this thesis, and we show the evaluation and future work.

CHAPTER TWO: BACKGROUND / RELATED WORK

2.1 Introduction

In This chapter we will display the literature review and related work of our enhanced technique, we talk about the classification of data mining as a whole and this will be as follows.

2.2 Classification in Data Mining

In machine learning and statistics, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. The classification of data mining is divided into two-stages , in the first one, a classification algorithm is utilized to pick up a rule from training data set. The next stage includes utilizing the rules expanded in the previous stage to foretell level of examine object. Here we have to note that classification in datamining does not have to be rule based, but for this research and our targeted algorithm needs, we will only focus on rule-based classifiers.

2.2.1 One Simple Rule

This algorithm (One Rule) 1R is the easiest Classification algorithm by (R.C. Holte, 1989). Which constructs a one-level decision tree and derives rules for training instances associated with most frequent classes. There are many issues for classification algorithms that were raised, the main two issues are lost attribute values and actual value attribute. Experiential researches explained that, in many classification situations, an easy approach like 1R creates rationally accurate classifiers.

2.2.2 Divide and Conquer approaches

The establishment of "divide and conquer" approach starts by selecting a root node, then this approach creates a branch for every stage of that node as much as possible. This indicates to divide the whole training data into sub groups, one for every value of the node. The similar manner will be recalled repeatedly until all data cases that belong to specific branch with also the similar class or the remaining data cases cannot be divided again. All nodes that connect between the roots to the leaf named intermediate nodes.

There are many different algorithms that utilize this approach in discovering the knowledge like C4.5 (Quinlan, J.R, et.al, 1993)

2.2.2.1 Decision Trees

Decision Trees model (Quinlan, J.R, 1979. 1986.) is the most famous one for classification and prediction. In this model structure, the candidate row logs the root node and the branch of every value that may be for the candidate is built. This procedure will stratified again in order to make all rows in a node terminate in the similar class or until the tree reaches the last level where no division can occur again. When the tree has been built, every route from the root to every branch (reach each leaf) produced a rule. The form of the rule is described by the route from the root to the branch node, and the resultant is the main class that is mentioned by the branch node.

Different pruning methods are utilized to abstract the rule from unimportant ones. This method may occur either by exchanging some sub-trees with branch nodes, or elevating a node to exchange the node at the top of the tree(Quinlan, J.R,1993). These two models are types of post-pruning tree approaches. Another efficient pruning model is to respect the error average at the inner branch nodes, then make a comparison between the averages of errors for the nodes with their exchange leaves (Quinlan, J.R, 1987).

2.2.2.2 ID3 Algorithm:

(Quinlan, J.R, 1979) defined ID3 algorithm as an approach that employs statistical features named information gain, to estimate the suitable feature used in a decision node. This algorithm chooses the root node depending on the feature that supplies much information than others, the procedure of the feature election is made again at the so-called child nodes of the root, except any feature that was elected before, while the remaining training data cannot divide again. Information gain calculates the range that the given feature splits the training data parts into classes.

The general ID3 algorithm is altered to maintain missing value and continuous attributes. As well various pruning models identified to result a minimal subsets of rules like exchanging a sub-tree by a leaf node. This exchange happens when the average of the predicted mistake in the sub-tree is larger than in the branch node (leaf).

2.2.2.3 C4.5 Algorithm

Another algorithm which is expanded of the previous algorithm is C4.5 algorithm which was defined by (Quinlan, J.R, 1993), calculates for absent values, continuous attributes and pruning of decision trees. Such version that gathers such small alteration to C4.5 called "C5" was enhanced by (Quinlan, J.R, 2009).

The C4.5 algorithm handled the absent values by using possibility which was determined depending on the frequencies of various values for any attribute at a specific node in the decision tree. Continuous attributes are discretized by a discretization technique. The main advantage over the ID3 algorithm for the C4.5 is pruning. In C4.5 algorithm various two kind of pruning techniques were utilized: sub-tree exchange and pessimistic error assessment in order to abstract the structure of the decision tree (L. Breiman, et al, 1984) (Quinlan, J.R, 1987), sub-tree exchange is possible to be done if a predicted error is greater than its exchange leaf. In this situation, the decision tree will be clipped by exchanging all the sub-tree by a branch node. J48 is an enforcement of C4.5 under the WEKA data mining platform.

2.2.3 Statistical Approach (Naïve Bayes):

In this approach the statistical design (R.C. Holte, 1993) differentiate from the 1R algorithm. It utilizes every existing feature to create a forecast. Such of this famous algorithm is Naïve Bayes (R.O. and P.E., 1973). It determines the possibility of every grade of data object by the common possibilities of feature values in that data object set. It considers the possibility of data object constraint is separate from the other data object possibilities in such class. This algorithms assumption is very hopeful that features in an actual environment data set are related with others and may have various levels of significance. Naïve Bayes was demonstrated to act good in different experimental researches.

2.2.4 Separate-and Conquer approach:

The Separate-and-Conquer approach starts by creating the rule in greedy model. Then, after a rule is established, all data cases wrapped by the rule will be rejected and this

manner is rejected again until the optimal rule created has a big error rate. Since in classification rules, there is just a one pre-identified class. There are many algorithms that utilized this model in discovering the rules such as PRISM (Cendrowska, 1987), RIPPER (Cohen w.w, 1995) and IREP (Furnkranz and Widmer, 1994).

2.2.4.1 Covering Approaches

This technique (Furnkranz, 1996) is a rule creation method for every class then making tests on the rule until the subset of cases covered by that rule are "pure". Then all cases covered by the rule will be rejected from any further processing, since the rule creation stage still happen until no other unclassified cases are left in the data sets. Building the classification rules are divided into direct model and an indirect model.

Direct models are those that extract rules immediately from data such as RIPPER. Indirect models are those that extract rules from other classification designs such as decision tree and C4.5. There are many classifiers concluded from these models such as PRISM, RIPPER and IREP.

Since the advantages of covering system is time efficiency in the process of establishing the rule immediately without inducing an intermediate decision tree as well as it directly rejects cases covered by the new rule from further induction.

2.2.4.1.1 Incremental Reduced Error Pruning

In (Furnkranz. and Widmer, 1994) a studying algorithm was defined named Reduced Error Pruning (IREP). It merges a separate-and-conquer model with Reduced Error Pruning (REP). REP model was defined as a way that clips and results a little set of division rules efficiently. IREP build a rule group in a greedy manner, the training data is divided into an increasing group and a clipping set in a random manner, where the increasing set includes 66.6% of the training data objects. Rules are created greedily in IREP, beginning from empty rule; a condition (attribute value) is attached to its former. Foil-gain measure (Quinlan, J.R and R.M. 1993) is used to select the suitable constraint to add and execute. IREP constantly combines constraints that increases the value of Foil-gain to the existing rule till to the rule covers no data objects from the rising group. When the rule build is completed, IREP rapidly construct clipping it inverses by deleting the

complete sequence of constraints from it. Beginning from the final constraint for each created rule, IREP construct deleting one constraint at a time and selects the deletion that enhances the confirmation function. An experimental research on various benchmarks exposed that IREP is quicker than REP and competitive to it with indication to error rate. Compared to C4.5 algorithm on 36 data sets, IREP obtained small error rate on 16 than others, while C4.5 was more efficient than IREP.

2.2.4.1.2 Repeated Incremental Pruning to Produce Error Reduction (RIPPER)

Repeated Incremental Pruning to Produce Error Reduction algorithm (RIPPER) was improved by (Cohen w.w, 1995). It creates the class of rules in the following steps: firstly, it splits the whole training data set into two different sets, a pruning set and a growing set. RIPPER considers the categorical via the previous two sets by frequently introducing rules beginning from empty rule group. The rule growing algorithm begins with empty rule, and experimentally combines just a single constraint at a unit of time while the rule prune of any error on the rising set.

RIPPER finished combining a rule by the minimum description length principle (MDL) when the rule is integrated, the whole length of rule characterization and the training data is evaluated. If the length of the characterization is longer than the shortest MDL gained so far, RIPPER stops combining rules. The MDL hypotheses states that the optimal design (set of rules) of data is the design which decreases the size of the model in addition to the amount of needed data to match the exclusion proportional to the design (I.H. Witten and E. Frank, 2000).

2.2.4.1.3 Prism

PRISM is a direct algorithm, which was introduced by (Cendrowska, 1987), the main goal of this algorithm is to make an immediate simulation to sorting rules from a training set. Each rule is generated by the algorithm by creating it term-by-term and choosing the attribute-value pair that increases the rule accuracy of a selection outcome class. Each term is defined in an "attribute = value" form.

In the mainly format, the prism algorithm uses separate-and- conquer technique, there are a specific steps started with determine the probability of each classification for all pairs, then choosing the highest probability pair and initiates a subset in order to create the training set, after that repeats the previous steps until having a subset contains only instance of classification.

Basically, PRISM adds tests to the condition of the rule, to get a maximum number of instances covered as well as to arrive 100% accuracy (higher accuracy or probability). The accuracy of the test is measured by (p/t) where (p/t) is a ratio of the number of positive instances (p)to the total number of instances covered by the rule (attribute being used) , after that the positive instances covered by the new rule are deleted from the data set for further rule generation.

2.2.4.1.3.1 TCS Prism

In (Bramer M, 2002), another copy of Prism was introduced and called TCS (Prism with Target Class Smallest first), which was defined to provide little groups of classification rules unlike the original form of the algorithm, with a same level of possible accuracy (probability) that may be achieved. In that form, the training set is prefixed to its base state before the rules are created for every class, thus the full training group needs to be processed once for every class.

1. Discover the class with smallest number of instances in the training set cases. Call this the target class TC.
2. Compute the probability of that class = TC for each possible attribute value pair.
3. Choose the pair of attribute/value that has with the highest probability and generate a group of the training set included all cases with the chosen migration
4. Go through 2 and 3 again for this subset till it contains just one case of class TC. The induced rule is then the conjunction of all the attribute value pairs selected in creating this subset.
5. Delete all cases covered by this rule from the training group.
6. Go through 1 to5 until you reach to the statues that no cases are still in the training group.

2.2.4.1.3.2 Parallel Prism approaches

Another approach was introduced by (Stahl and Bramer, 2008), it was a system architecture of P-Prism using a blackboard server containing two partitions, the first partition is for presenting rule terms to the blackboard (Local Rule Term Partition) and the second partition is to announce global information (global information partition) to the worker machines. The moderator program on the blackboard derives the global information, see (Figure 7).

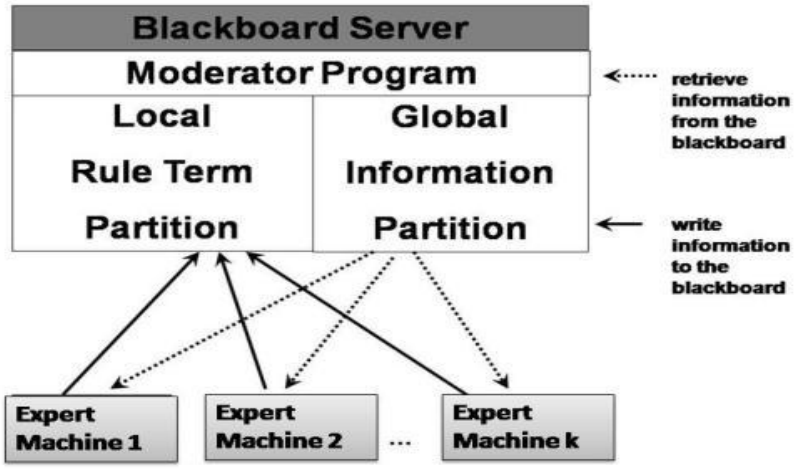

Figure 7 System architecture of the P-Prism algorithm(Stahl and Bramer, 2008),

Another enhanced module was introduced by (Stahl and Bramer, et.al, 2009). A Parallel Modular Classification Rule Induction (PMCRI) algorithm, it is based on the Cooperating Data Mining Model (CDM) that was introduced by (Provost, 2000). This is illustrated in (Figure8), the PMCRI algorithm applies to the CDM model and uses distributed blackboard System in its second layer for CDM model.

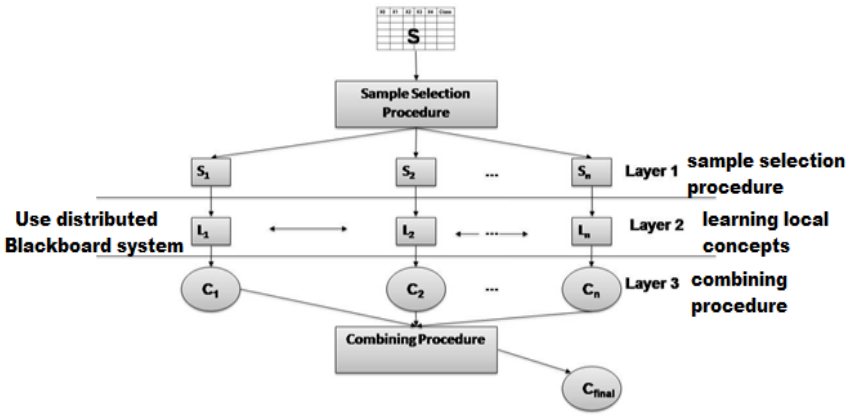

Figure 8 Cooperating Data Mining (Stahl and Bramer, 2008)

2.2.4.1.3 .3 Discretization of Continuous Attributes in prism

As we mentioned previously that the traditional prism cannot handle continuous attributes، but some of the other research that was mentioned can handle this data by using Chi-merge. The Chi-Merge algorithm consists of an initialization step and a bottom-up merging process, where intervals are continuously merged until a termination condition is met. Chi-Merge is initialized by first sorting the training data according to their value for the attribute being discretized and then constructing the initial discretization, in which each data is put into its own interval. The disadvantage of this method is the highest complexity which is equal to O (n log n), compared to other discretization approaches.

In (M.A Bramer, 2005) the TDIDT (Top-Down Induction of Decision Trees) and the Prism algorithm as implemented in Inducer, both have a facility for local discretization of continuous attributes, i.e. dividing the values of an attribute X into two parts, X<a and X>=a, at each stage of the rule generation process. However, many other rule induction algorithms have no facilities for dealing (directly) with continuous attributes and for purposes of comparison it is sometimes helpful for the user to be able to 'turn off' such attributes, effectively treating them as if they were specified as ignore attributes in the name file.

2.2.4.1.3 .4 Pruning prism Algorithms

2.2.4.1.3 .4 .1 J-measure

J-measure algorithm was defined by (P.smyth,et al,1991). They assured the importance of the J-measure as a quantity indicator of measuring the rule content of the data. The j-measure, also named the *cross-entropy*, is produced according to the following relation:

$$J(X; Y = y) = p(x \mid y) \cdot log2(p(x \mid y)p(x))+(1- p(x \mid y)) \cdot$$

$$log2\ ((1- p(x \mid y))\ (1- p(x))).$$

The value of *cross-entropy* related to two different values (Liu, Alexander and Frederic, 2013)

- P(x): which indicates the possibility that the outcome of the rule will be correspond if there is no other data specified. This is called a priori possibility of the rule outcome.
- *p(x|y)*: the possibility that the outcomes of the rule correspond if the specific antecedents are accepted. This is also read as *a posterior* possibility of *x* given *y*.

2.2.4.1.3 .4 .2 J-pruning

J-pruning, based on the J-measure which was mentioned previously, is a pre-pruning technique, because the pruning job is taken through rule creation process. It was improved by (Bramer, 2002).

J-pruning produced comparatively fine outputs as mentioned in (Bramer, 2002). However, Stahl and Bramer marked out in (Stahl and M.A. Bramer,2012) and (Stahl and M.A. Bramer, 2011) that this algorithm does not achieve the J-measure to its whole possibility, as this technique directly stops the creation process once the J-measure decreases after a new term is combined to the rule. In fact, it is probable that the J-measure may be decreasing and increasing frequently after further terms are combined to the rule. This predicted that the pruning job may be done much sooner.

The results show that J-pruning can obtain comparatively fine outputs that might be expressed by the supposition that it does not occur many times which explain that the J-value decrease and then increase many times. It also mentions that J-pruning may even produce under fitting rules due to over pruning rules. The reason of this, is that the pruning job may be taken much sooner output coming in much public rules created to have high possible accuracy. This induced the enhancement of a new pruning technique, named J-max pruning, which was defined by (Stahl and. Bramer, 2011) to exploit the J-measure to its whole potential.

2.2.4.1.3 .4 .3 J-max pruning

As produced previously, J-max pruning may be visible as a combination between pre-pruning and post pruning. But, with consideration to every generated rule, every individual rule is really post-pruned after the perfecting of the creation for that rule.

2.2.4.1.3 .4 .4 J-mid pruning

Inducing from the previously mentioned algorithms, some researchers (Han Liu, et al, 2013) suggested a novel pruning method that not only minimizes over fitting of classification rules but also avoids under fitting and unneeded rule creations and avoids their related computing cost.

A novel pruning algorithm named J-mid pruning that is depending on the J-measure is defined and demonstrated in this work. The practical research shows that this algorithm can avoid under fitting and unnecessary computing overheads and minimizing over fitting of classification rules. In most cases, J-mid pruning produces the Prism method that creates a rule group with the same level of complication with J-pruning and J-max pruning algorithms. On the other hand, it is possible that J-mid pruning also causes Prism to create smaller but more general rules than J-pruning or J-max pruning. In addition, in some special cases, J-mid pruning completes rule creation faster than J-max pruning. This prevents repeated effort in deleting terms subsequently from a rule. So, the authors here confirmed that the J-mid pruning technique using more datasets in terms of the amount of outcast rules.

2.2.5 Hybrid Approach

The PART algorithm differentiate about C4.5, it is divided into different stages, PART algorithm creates a one rule at a time unit by avoiding comprehensive pruning (E. Frank, et al, 1998). PART appoints separate-and-conquer to create a group of rules and utilizes divide-and-conquer to create resolution tree and create fractional resolution trees as in C4.5.

Every rule in PART algorithm matched with the leaf node in the fractional resolution tree with the largest cover-up. Lost values and clipping approaches are handled in a similar manner in C4.5.

Empirical check utilizing PART, RIPPER and C4.5 on various data set have been determined in (E. Frank, et al, 1998). The outputs detected that regardless of the straight forwardness of PART algorithm, it creates groups of rules and it is the accurate algorithm comparing with C4.5 and RIPPER algorithms.

CHAPTER THREE: RESEARCH METHODOLOGY

3.1 Introduction

One of the constraints that limits the potentials of Prism, is that Prism cannot deal with continuous data, but a lot of attributes are actually continuous attributes, such as currency fields or years of experience fields or any numeric attributes is by default continuous, so to handle these cases by Prism some researchers suggested using discretization methods as a pre-processing step for prism, one paper suggested using a well-known discretization method called chi-merge (Kerber, 1992), but the problem with chi-merge is simply the high calculations needed to discretize the data. In data mining most of the studied data is considered to be Big-data so high calculation ratio is not recommended when dealing with it, so to solve this drawback other methods will be used in this research to discretize continuous attributes for the Prism algorithm as a pre-processing step.

3.2 Handling continuous attributes in prism algorithm

One discretization method we use is the Equal-frequency method, we choose this method because it has an advantage over other unsupervised methods such as the Equal-width method, in the Equal-width method intervals are fixed to a certain size regardless of how much data they contain, this is not efficient because some intervals will be heavy loaded with data whereas other intervals will be left totally empty.

Supervised discretization methods are methods that determines data intervals for a continuous attribute depending on the associated targeted class attribute, whereas unsupervised methods does not depend at all on the class attribute.

In addition, the Equal-frequency method got an advantage over supervised discretization methods such as the Entropy method, one is that we are faced with high calculation complexity which as previously mentioned is not preferred when working with big-data, also intervals or the number of sets is not specified by the user but determined by the class type and data.

Nevertheless, using a supervised method such as Entropy is recommended when we have a well-sorted class attribute, where this leads to less calculations and intervals, while using unsupervised methods such as the Equal-frequency method is said to be good and recommended when the targeted class is very diverse and no certain sorting method is used.

3.3 Using other discretization methods

We used discretization pre-processing methods for the Prism algorithm that was not used before in order to inspect if better results can be obtained regarding time and space complexity measures.

For the continuous data not previously handled by prism we added a preprocessing discretization step to handle this kind of data, regarding the fact that prism depends on a targeted class then the discretization method type should consider this, so if the targeted class values are highly variant and unsorted then an unsupervised method is used (equal-frequency), otherwise a supervised method is used (Entropy), these methods are not new but integrating them with Prism was not done before.

3.4 Enhanced Prism algorithm (E-prism)

The Prism algorithm suffers from redundancy in its rule generation phase, this is considered wasted time regarding the fact that this redundancy can be avoided.

Another problem to be searched, is the problem of handling continuous data, the traditional Prism cannot handle this type of data, and some approaches were implemented to solve this, in this thesis we will try to use different approaches to get better results.

Another main contribution in this research is that the traditional Prism algorithm is suffering from redundancy in one of its main phases which is the (rule generation phase).

As mentioned before how Prism works, it mainly detects the strongest attribute sets with a desired class attribute by the user, depending on that, a rule is generated to help in decision making for future data.

During this step if Prism detects one or more equal relation strength values it does not handle this case with different handling methods, it just re-runs the whole rule generation cycle again, this is not necessary, since by experiment and by logic researchers proved that the left behind equal attribute sets will be chosen for sure in the next iteration, so why doing the calculation then the filtering then choosing the values that is already known to be chosen since they have the highest relation ratio with the targeted class.

What this research is implementing, is breaking the unnecessary redundant steps of re-filtering data and recalculating the attribute sets, using a fixed rule that indicates the following, if equal relations are detected then one of them is randomly (arbitrarily) added to the generated rule, and the other is directly added after it, bypassing the unnecessary filtering and calculations, this will result to an exact generated rule as in the traditional prism without the redundant overhead.

This enhancement on the Prism algorithm is very effective especially when we are handling big data, in data mining, the typical case is having big data, so any enhancement in this area will be valuable to reduce time and space complexity.

3.4.1 Enhancing the rule generation process complexity

In this research we enhanced the rule generation phase in the Prism algorithm, the latest version of Prism works on finding the highest accuracy (probability) value for each item, then adds this selected item to the rule, the problem is that when accuracy (probability) vales are equal, Prism chooses one of them randomly (arbitrarily) or based on a predetermined value, the other value will be obviously selected in the next filtering round because as we said before it has the same accuracy (probability) value as the previously selected item, so to solve this redundancy, we rebuild the rule generation choosing step, to make it process both items in the same iteration, saving the time needed to recalculate all other items again.

Mainly what we did in the generation phase, we removed the wasted time derived from redundancy caused by items having equal probability, we detected that the algorithm recalculates after choosing one of the equal values, despite the fact that it will always choose the same value, so we modified this step to let it choose both of the items without recalculating and filtering data and probabilities all over again.

In this research, we aim to enhance the prism algorithm by adding a pre-processing discretization step and a modified rule generation phase. The E-prism algorithm block diagram is shown in (Figure 9).

Also in (chapter 5) we will provide a step by step demonstration for the E-Prism algorithm working on a benchmark dataset, this will show how E-Prism works in the real environment.

3.5 The data flow diagram for the proposed algorithm

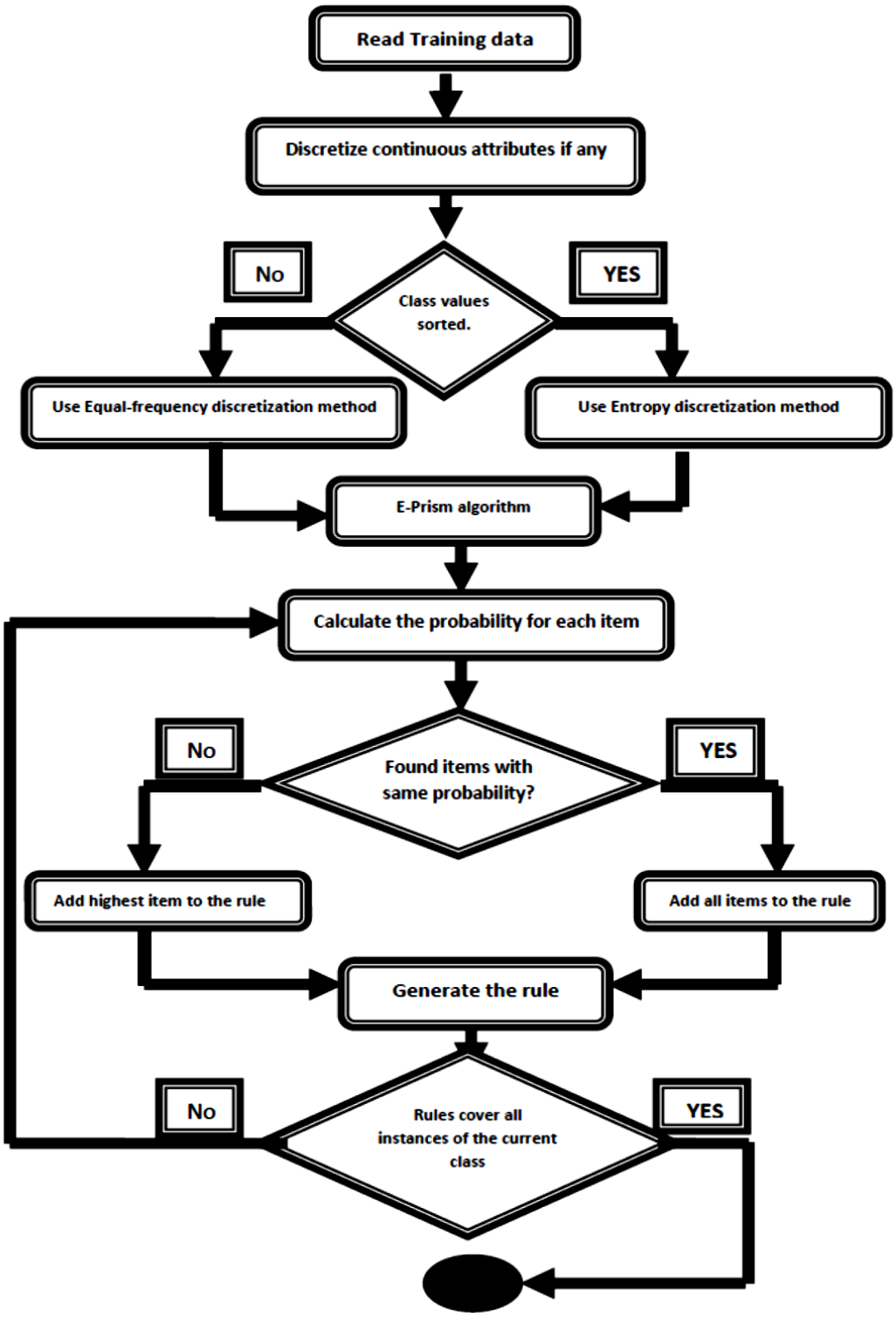

Figure 9 The data flow diagram for the proposed algorithm

CHAPTER FOUR: IMPLEMENTATION DETAILS AND FEATURES

4.1 Introduction

In order to describe our approach in a formal manner we will use the Z-notation formal specification language, Z language was described in (Jacky, J, 1997) as being a widely used for describing and modeling computing systems. It is targeted at the clear specification of computer programs and computer-based systems in general.

4.2 Formalize the contribution using Z notation

Z language is a group of notations used to present mathematical text, it uses simple mathematics to describe systems, programs and algorithms. Z is used to model hardware as well as software.

Z does not restrict you regarding what you can model; it is very wide and scalable. Z is just a notation language, cannot call it a method; Z notation can support many different methods. The meaning of a Z text is determined by its authors. It can be understood to model only the behavior of a system. Z text can be understood to represent blocks and parts of a code: modules, data types, procedures, functions, classes, objects. Using other words, Z model is a detailed formal version of a system.

In the paper of (VOTING R. O., 2002), details and standards of the Z specification language were introduced, explained and discussed.

In this section we will demonstrate our enhancement on the E-Prism and compare it with the previous traditional prism (T-Prism), we simulated the scenario of having two items having the same probability, as we mentioned before the original prism does not handle this, it just recalculate and filter again, in our work it takes both items and add them to the rule, saving time and calculation effort.

4.2.1 Prism schema (state space)

Types of the specification: [Att, Items, Rule, Classes, IT], the first feature of the system to be described is its state space, and we do this with a schema:

Prism

$att : Att$
$item : Items$
$class : \mathbb{P}\ Classes$
$Classes : att \nrightarrow item$
$R : Rule$
$It : IT$

$Item = \text{dom}\ class$
$Class \subseteq Classes$
$Item1 \subseteq Items$
$Item2 \subseteq Items$
$It1 \subseteq IT$
$It2 \subseteq IT$

In this schema we defined variables, it will be a description for each variable:

➔In the Prism schema:

att is an attribute type of Att

Item is type of Items

Class is set of classes

Classes is an attribute that has a relationship with Items

R is type of Rule

It is iteration type of IT

➔Predicate in the Prism schema:

Item is domain class

Class is sub set of classes

Item1 is sub set of items

Item2 is sub set of items

It1 is sub set of IT

It2 is sub set of IT

4.2.2 Probability schema

This schema is an operation schema to calculate the probability for both items, through calculating the item occurrence with its class and its occurrence with the targeted class.

Probability

Ξ Prism $P : \mathbb{R}$ $N : \mathbb{Z}$
$N1 \subseteq N$ $N2 \subseteq N$ $N3 \subseteq N$ $N4 \subseteq N$ $P1 \subseteq P$ $P2 \subseteq P$ $N1 = \#(item1 \in class)$ $N2 = \#(item1 \in classes)$ $N3 = \#(item2 \in class)$ $N4 = \#(item2 \in classes)$ P 1= N1/N2 P 2 = N3/N4

The following is a description of each variable mentioned above:

Ξ read from prism

P is real number

N is integer.

➔Predicate in Probability

N1 is sub of N, and it is the number of item that belongs to class.

N2 is sub set of N, and it is the number of item that belongs to another classes.

N3 is sub set of N, and it is the number of item that belongs to class.

N4 is sub set of N, and it is the number of item that belongs to another classes.

P1 is a sub set of P, and it represents the probability Presence item with class (N1) divided over the probability Presence item with another classes (N2).

P2 sub of P, and it represents the probability Presence item with class (N3) divided over the probability Presence item with other classes (N4).

4.2.3 Schema T-Prism (traditional prism)

The next schema is for the traditional prism, it will be used to illustrate the redundancy in rule the generation phase.

T- Prism

Ξ Prism
item*?:Items*
R!:Rule

(P 1= *P* 2)
It1⇒R= **if** item1∨ item2**then** class1
It2⇒R= **if** item1∧ item2**then** class1

Item? : is input

Rule! : is out put

In the first iteration If probability item1 = probability itme2 then the traditional algorithm will select one of them randomly and add it to Rule, in the second iteration Prism will select item2 and add it to the Rule, Figure 10 is a Pseudocode for T-Prism.

```
For each class C
  Initialize E to the instance set
  While E contains instances in class C
    Create a rule R with an empty left-hand side that predicts class C
    Until R is perfect (or there are no more attributes to use) do
      For each attribute A not mentioned in R, and each value v,
        Consider adding the condition A = v to the left-hand side of R
        Select A and v to maximize the accuracy p/t
          (Break ties by choosing the condition with the largest p)
      Add A = v to R
    Remove the instances covered by R from E
```

Figure 10 Pseudocode for T-Prism

4.2.4 Schema E-Prism (Enhanced prism)

The next schema is for the enhanced prism, it will be used to illustrate how the redundancy in rule generation phase was solved.

E- Prism

Ξ Prism
item*?:Items*
R!:Rule

(P 1= *P* 2)
It1⇒R= **if** item1∧ item2 **then** class1

Item? : means item is input.

Rule! : means Rule is out put

In the first iteration If probability item1 = probability itme2 then the enhanced Prism algorithm will select both items and add them to the rule. This enhance will reduce time and some unnecessary overhead, Figure 11 is the Pseudocode for E-Prism.

```
For each class C
  Initialize E to the instance set
  While E contains instances in class C
    Create a rule R with an empty left-hand side that predicts class C
    Until R is perfect (or there are no more attributes to use) do
      For each attribute A not mentioned in R, and each value v,
        Consider adding the condition A = v to the left-hand side of R
        Select A and v to maximize the accuracy p/t
           (break ties by choosing both conditions)
           Add Ai = vi to Ri = number of equal maximum attributes
    Remove the instances covered by R from E
```

Figure 11 Pseudocode for E-Prism

4.3 Evaluate contribution by calculating complexity

➔ In the traditional prism the complexity was: O (n.m.x.s)

Where: n is number of distinct attributes.

m: is number of distinct items.

x: is size of table.

s: is number of iteration.

➔ In our algorithm (E-prism) the difference will be in the number of iteration(s).

The number of iterations (s), will inversely proportional with equal elements.

CHAPTER FIVE: RESULTS AND EVALUATION

.

5.1 Introduction

We compare our modified algorithm with the traditional prism algorithm and other rule induction approaches using the benchmark data-sets provided by the Weka data mining tool, where we will compare the processing time of our modified algorithm with the processing time of other algorithms provided by the Weka tool.

5.2 Case study and Evaluation

Here is a simple case study on PRISM algorithm. Assume that we want to derive a rule for "recommendation = hard" this rule will be derived based on the following dataset that belongs to the well-known Weka "lenses dataset".

Table 2: Dataset for contact-lenses (weka tool)

#	Age	prescription	Astigmatism	Tear rate	Recommendation
1	young	myope	no	Reduced	none
2	young	myope	no	Normal	soft
3	young	myope	yes	Reduced	none
4	young	myope	yes	Normal	hard
5	young	hypermetrope	no	Reduced	none
6	young	hypermetrope	no	Normal	soft
7	young	hypermetrope	yes	Reduced	none
8	young	hypermetrope	yes	Normal	hard
9	pre-presbyopic	myope	no	Reduced	none
10	pre-presbyopic	myope	no	Normal	soft
11	pre-presbyopic	myope	yes	Reduced	none
12	pre-presbyopic	myope	yes	Normal	hard
13	pre-presbyopic	hypermetrope	no	Reduced	none
14	pre-presbyopic	hypermetrope	no	Normal	soft
15	pre-presbyopic	hypermetrope	yes	Reduced	none
16	pre-presbyopic	hypermetrope	yes	normal	none
17	presbyopic	myope	no	reduced	none
18	presbyopic	myope	no	normal	none
19	presbyopic	myope	yes	reduced	none
20	presbyopic	myope	yes	normal	hard
21	presbyopic	hypermetrope	no	reduced	none
22	presbyopic	hypermetrope	no	normal	soft
23	presbyopic	hypermetrope	yes	reduced	none
24	presbyopic	hypermetrope	yes	normal	none

Next is presented all the candidate tests and their accuracies after choosing the "recommendation = hard" as a class label.

As shown by (Figure 12) there are two attribute values having the same probability, that is 4/12, in this case the traditional Prism tends to take one value and adds it to the rule, then recalculate and filter one more time while always having the previous equal value, in our algorithm both values are added to the rule directly, without re-calculating and re-filtering, this reduces redundancy that used to cause unnecessary overhead for Prism.

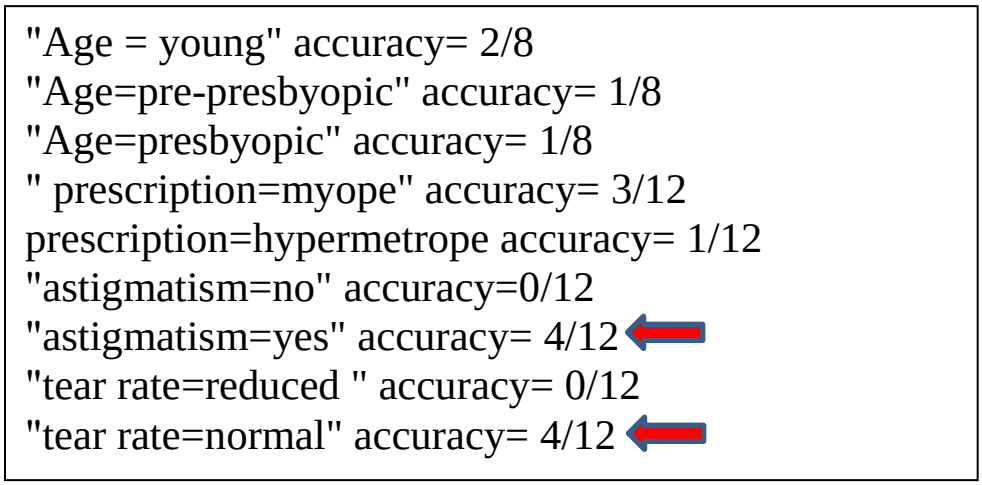

Figure 12 First iteration results

The traditional Prism selects one of the equal probabilities randomly, let's assume that it takes "astigmatism = yes".

Then the first rule will be "If astigmatism = yes then recommendation = hard".

Now, consider the remaining possible tests in order to refine the rule.

The subset of the training set covered by this incomplete rule is given in (Table3).

Table 3: Dataset after first iteration filtering

#	Age	prescription	astigmatism	Tear rate	Recommendation
1	young	myope	yes	reduced	none
2	young	myope	yes	normal	hard
3	young	hypermetrope	yes	reduced	none
4	young	hypermetrope	yes	normal	hard
5	pre-presbyopic	myope	yes	reduced	none

6	pre-presbyopic	myope	yes	normal	hard
7	pre-presbyopic	hypermetrope	yes	reduced	none
8	pre-presbyopic	hypermetrope	yes	normal	none
9	presbyopic	myope	yes	reduced	none
10	presbyopic	myope	yes	normal	hard
11	presbyopic	hypermetrope	yes	reduced	none
12	presbyopic	hypermetrope	yes	normal	none

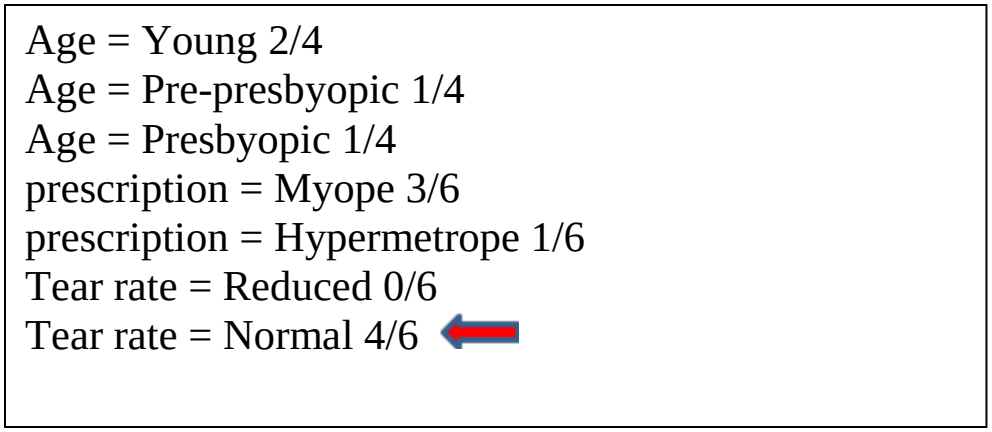

Figure 13 Second iteration results

The highest value in the second iteration (figure13) was "Tear rate = Normal" with accuracy=4/6, which appeared but was neglected in the first iteration (figure12) as the highest value with “astigmatism = yes". But traditional prism selected “astigmatism = yes" randomly, in this iteration it will add "Tear rate = Normal" to the rule. Now the rule is "If astigmatism = yes and tear production rate = normal then recommendation = hard". In our algorithm both values are added to the rule directly from the first iteration, without re-calculating and re-filtering, this reduces the redundancy level in Prism.

The subset of the training set covered by this incomplete rule is given in (table4).

Table 4: Dataset after second iteration filtering

#	Age	prescription	astigmatism	Tear rate	Recommendation
1	young	myope	yes	normal	hard
2	young	hypermetrope	yes	normal	hard
3	pre-presbyopic	myope	yes	normal	hard
4	pre-presbyopic	hypermetrope	yes	normal	none
5	presbyopic	myope	yes	normal	hard
6	presbyopic	hypermetrope	yes	normal	None

Age = Young 2/2
Age = Pre-presbyopic 1/2
Age = Presbyopic 1/2
prescription = Myope 3/3
prescription = Hypermetrope 1/3

Figure 14 Third iteration results

At this moment, Prism stops, it reached its stop condition which is having 100% accuracy for the current class.

5.3 Evaluation and results

To get results that prove the efficiency of our algorithm, we have programed the Traditional Prism and the E-Prism algorithms using the C# programming language, then using timestamp variables placed on the two ends of the E-Prism code, we detected the run-time for E-Prism. Then we processed different benchmark datasets using this code, detecting the different timings the code took to generate rules for each dataset. To maintain accuracy we run the code on the same dataset multiple times (ten times for each dataset) to detect the mean average for its time to generate rule.

We used the following benchmark datasets to compare results: Contact-lenses, Weather numeric and Weather nominal official Weka datasets, these datasets can be found on the Weka tool online portal. Weather numeric data set cannot be handled using traditional prism, in our algorithm (E-prism) we can handle this type of data sets by using (Equal-frequency and Entropy) techniques.

In (Table 5) a detailed comparison is shown for three rule induction algorithms, they are: Traditional prism algorithm, Ripper (JRIP) algorithm and our enhanced E-prism algorithm.

Table 5 Rule generation time complexity simulation

Data set by Weka tool	**Algorithm used**	**Time to generate rule**
contact-lenses	Traditional prism algorithm in weka	0.01 seconds
Weather numeric	Traditional prism algorithm in weka	0.01 seconds
Weather nominal	Traditional prism algorithm in weka	0.01 seconds
contact-lenses	Traditional prism algorithm in C#	0.02 seconds
Weather numeric	Traditional prism algorithm in C#	0.015 seconds
Weather nominal	Traditional prism algorithm in C#	0.012 seconds
contact-lenses	Ripper(JRIP) algorithm in weka	0.01 seconds
Weather numeric	Ripper(JRIP) algorithm in weka	0.01 seconds
Weather nominal	Ripper(JRIP) algorithm in weka	0.02 seconds
contact-lenses	E-prism algorithm in C#	0.009 seconds
Weather numeric	E-prism algorithm in C#	0.007 seconds
Weather nominal	E-prism algorithm in C#	0.006 seconds

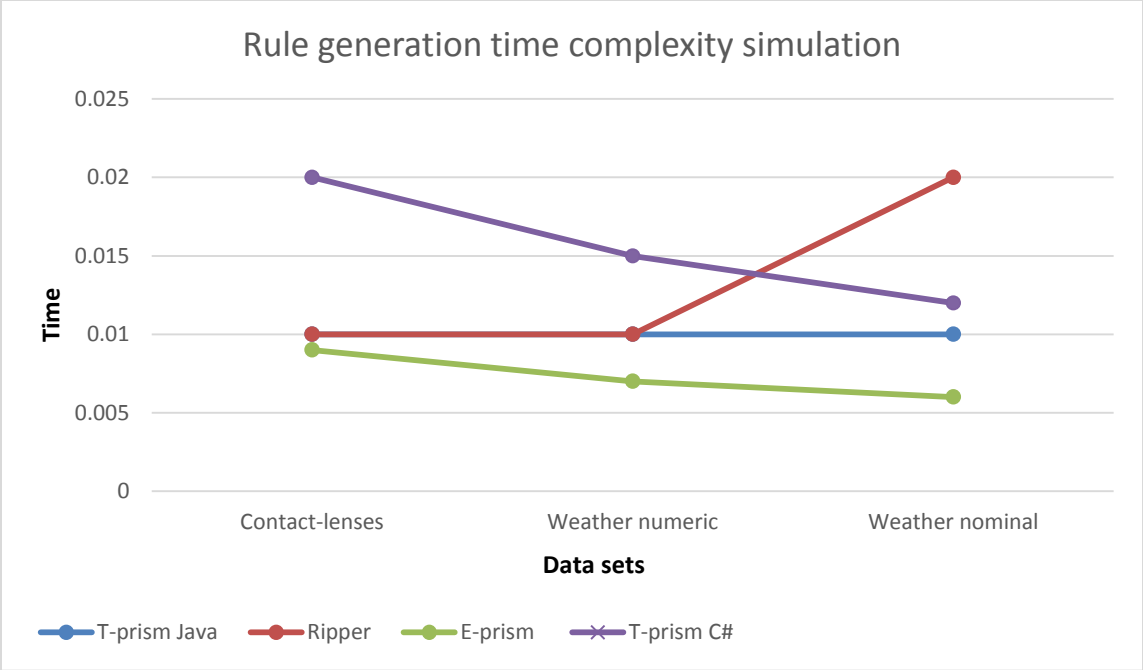

Figure 15 Rule generation time complexity simulation

5.4 Implementation Issues / Application areas

In this study, we are working in the data mining field, this field is well known to be as a must have in most domains, especially those with big data transfer amounts that is used to help in decision making. In this sense, data mining aids businesses with approaches and techniques that utilize historical data to build future rules, decisions and company future plans. We can implement our work on various data sets that have been previously

tested with other similar algorithms, by this way we will be able to detect the enhancement provided by our algorithm.

We expect that we can use our E-Prism algorithm in the following application domains:

- Forecast prediction: weather news.
- Sales and marketing: Market behavior based on buying patterns.
- Financial domains: credit card issuing, prediction of finance investments.
- Health care/Medicine: detect long-run side effects of treatments on patients.

Chapter six: Conclusions and Future work

6.1 Conclusions

In this research we enhanced the phase of rule generation in the Prism algorithm, this was carried out in order to reduce processing time, which will be utilized in all Prism algorithm implementations in different domains.

We enhanced the rule generation phase by removing a duplicated unnecessary step, this resulted in great enhancement regarding the time complexity of the Prism algorithm.

Furthermore, we handled continuous data in prism algorithm using two discretization techniques (Equal-frequency and Entropy), we have proven that with Prism, choosing only supervised methods is not efficient nor using only unsupervised methods, this happens because in Prism we have high dependency on the class data. In this research we combined the usage of two different techniques, one is Equal-frequency from the unsupervised methods, and the other is Entropy from the supervised methods.

6.2 Open issues/directives for Future Work

For future work, we are aiming to enhance our system by merging it with parallel rule induction approaches using multiple machines in order to be capable of handling bigger datasets in the most efficient way.

Furthermore, we are planning to combine a pruning method to E-Prism in order to add a pruning phase to remove unnecessary rule over fitted rules.

REFERENCES

Bramer M., (2002). "An Information-Theoretic Approach to the Pre-pruning of Classification Rules", Proceedings of the IFIP Seventeenth World Computer Congress TC12, 201-212.

Bramer, M., (2005). "Inducer: a public domain workbench for data mining". International Journal of Systems Science, 36(14), 909-919.

Cendrowska.J. (1987). "PRISM: An algorithm for inducing modular rules". International Journal of Man-Machine Studies, Volume 27, Issue 4, pages 349-370.

Cohen. W. W. (1995.),"Fast effective rule induction". In the Proceeding of the 12th International Conference on Machine Learning, pp. 115-123, Morgan Kaufmann..

Daniela Joita, (2010),"Unsupervised Static Discretization Methods in Data Mining", Revista Mega Byte, vol. 9, 2010.

Dougherty, J. Dougherty, R. Kohavi, M. Sahami. (1995), "Supervised and unsupervised discretization of continuous features". In Proceedings of the 12th International Conference on Machine Learning (1995), pp. 194-202.

E Xu, Shao Liangshan, Ren Yongchang, Wu Hao and Qiu Feng, (2010) "A new Discretization approach of Continuous attributes", Asia-Pacific Conference on Wearable Computing Systems, vol. 5, no. 2, pp. 136-138, 2010.

E. Frank and I.H. Witten (1998), "Generating accurate rule sets without global optimization," Proc 15th International Conf on Machine Learning, 1998, pp. 144-151.

Fayyad, Usama M.; Irani, Keki B. (1993) "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning", Proceedings of the International Joint Conference on Uncertainty in AI (Q334 .I571 1993), pp. 1022-1027.

Furnkranz. J and Widmer.G (1994)," Incremental reduced error pruning (IREP)", in Cohen.W and Hirsh (eds.), proceeding of the 11th international conference on machine learning (ML94), 70-77,Morgan Kaufmann,1994.

Furnkranz. J. (1996) ,"Separate-and-conquer rule learning". Technical Report TR-96-25, Austrian Research Institute for Artificial Intelligence, Vienna.

Han Liu, Alexander Gegov and Frederic Stahl, (2013), "J-measure Based Hybrid Pruning for Complexity Reduction in Classification Rules", Issue 9, Volume 12, September 2013, E-ISSN: 2224-2678.

I.H. Witten and E. Frank (2000), "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations", Morgan Kaufmann.

Jacky, J. (1997). “The way of Z: practical programming with formal methods”. Cambridge University Press.

Joao Gama and Carlos Pinto, (2006), “Discretization from Data Streams: Applications to Histograms and Data Mining”, Symposium on Applied Computing, pp. 662-667.

Kerber. Randy, (1992). “ChiMerge: Discretization of numeric attributes”, Proceedings of the tenth National Conference on Artificial Intelligence, pp.123-128, 1992.

L. Breiman, J.H., R.A. Olshen, and C.J. Stone (1984), Classification and Regression Trees, Wadsworth.

Llia Mitov, Krassimira Ivanova, Krassimir Markov,(2009), “comparison of discretization methods for preprocessing data for pyramidal growing network classification method”, international book series "information science and computing.

M. A Bramer, (2005), "Inducer: A Public Domain Workbench for Data Mining".

M.A. Bramer, (2002), “An information-theoretic approach to the pre-pruning of classification rules”, in: B.N. M Musen, R. Studer (Eds.) Intelligent Information Processing, Kluwer, 2002, pp. 201–212.

P. Smyth and R.M. Goodman, (1991) ,”Rule Induction Using Information Theory”. In: G. Piatetsky- Shapiro and W.J. Frawley (eds.), Knowledge Discovery in Databases. AAAI Press, pp. 159-176.

Provost F, (2000), "Distributed data mining: Scaling up and beyond", Advances in distributed and Parallel Knowledge Discovery", MIT Press, p. 3–27.

Quinlan, J.R, (1987) “Simplifying decision trees,” International Journal of Man-Machine Studies, vol. 27, Sep. 1987, p. 221–234.

Quinlan, J.R, (1979), “Discovering rules by induction from large collections of examples,” Expert Systems in the Microelectronic Age, D. Michie, ed., Edinburgh University Press, 1979, pp. 168-201.

Quinlan, J.R, (1986) “Induction of decision trees,” Machine Learning, vol. 1, Mar. 1986, pp. 81-106.

Quinlan, J.R, (1987), “Generating production rules from decision trees,” Proceedings of the 10th international joint conference on Artificial intelligence - Volume 1, San Francisco: Morgan Kaufmann Publishers Inc., 1987, p. 304–307.

Quinlan, J.R,, “Data Mining Tools See5 and C5.0,” vol. 2009, 2008

Quinlan, J.R. and R.M. (1993), “FOIL: A Midterm Report,” Machine Learning ECML93 European Conference on Machine Learning Proceedings, P. Brazdil, ed., Springer-Verlag, pp. 3-20.

Quinlan. J. R. (1993). C4.5: Programs for Machine Learning “Published Book”.

R.C. Holte, (1993), “Very Simple Classification Rules Perform Well on Most Commonly Used Datasets,” Mach. Learn., vol. 11, 1993, p. 63–90.

R.O. Duda and P.E. Hart, (1973), Pattern Classification and Scene Analysis, Wiley-Blackwell.

Rajashree D, Rajib L, Rasmita D,(2011)," Comparative Analysis of Supervised and Unsupervised Discretization Techniques", International Journal of Advances in Science and Technology, Vol. 2, No. 3, 2011.

Sellappan Palaniappan, Tan Kim Hong, “Discretization of Continuous Valued imensions in OLAP Data Cubes”, International Journal of Computer Science and Network Security, vol. 8, no. 11, pp. 116-126, 2009.

Sheng-yi Jiang, Xia Li, Qi Zheng and Lian-xi Wang, (2009) “An Approximate Equal Frequency Discretization method”, WRI Global Congress Intelligent System, vol. 3, no. 4, pp. 514-518, 2009.

Stahl and M.A. Bramer, (2012) “Jmax-pruning: A facility for the information theoretic pruning of modular classification rules”. Knowledge-Based Systems 29 12-19

Stahl and M.A. Bramer. (2011),” Induction of modular classification rules: using Jmaxpruning”. In Thirtieth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, 14-16 December 2011, Cambridge.

Stahl F. and Bramer M., (2008). "P-Prism: A Computationally Efficient Approach to Scaling up Classification Rule Induction", in IFIP International Conference on Artificial Intelligence, Milan.

Stahl F., Bramer M. and Adda M., (2009). "PMCRI: A parallel modular classification rule induction framework", Proceedings of the 6th International Conference, MLDM 2009, Leipzig, Germany, July 23-25.

VOTING, R. O. (2002). Information technology--Z formal specification notation Syntax, type system and semantics.

ملخص

في مجال علم الحاسوب يعرّف مصطلح التنقيب عن البيانات بأنه هو المجال الذي يعمل على ايجاد العلاقات في مجال معين وذلك باستخلاص هذه العلاقات من البيانات الموجودة مسبقاً والتي تسمى بالبيانات التجريبية. باستخدام هذه العلاقات يتم توليد القواعد التي ستساعد متخذي القرار على اتخاذ القرارات بناء على البيانات والمعلومات التي تم رصدها في نفس المجال سابقا.

أحد هذه الطرق هي طرق تغطية البيانات، حيث تقوم هذه الطريقة على مبدأ دراسة أكبر قدر ممكن من البيانات وتستخرج العلاقات منها حسب قوة ترابط هذه البيانات مع الهدف المحدد او القرار المعين الذي يريده المستخدم.

إحدى الخوارزميات المهمة التي تطبق هذه الطرق، هي خوارزمية برزم، هذه الخوارزمية تعتمد على مفهوم فرق تسد، حيث تقوم بتقسيم البيانات ودراستها وتصفيتها حتى تصل في النهاية إلى بناء قواعد تساعد صانع القرار على اتخاذ القرارات، وذلك بناء على دراسة البيانات التي تم جمعها مسبقا، لا بناء على رؤية قاصرة للبيانات الحالية.

تقوم خوارزمية برزم بتوليد القواعد من خلال اكتشاف العلاقات القوية بين العناصر والبيانات، المرحلة الرئيسية من مراحل هذه الخوارزمية هي عملية توليد القواعد، في هذه المرحلة يتم حساب الاحتمالية وقوة الارتباط لكل العناصر، حيث تُحسب هذه النسبة للاحتمالية من خلال تقسيم احتمالية ورود العنصر مع الصنف المستهدف في الدراسة على ورود نفس العنصر مع باقي الأصناف ومقارنتها، وعندما تظهر نتيجة الاحتمالية لكل العناصر، حينها يتم اخذ اعلى احتمالية وارتباط ليتم اضافة العنصر المرتبط بها الى القاعدة العامة.

حتى هذه النقطة لا يوجد مشاكل ظاهرة، ولكن من خلال المشاهدات تم ملاحظة عملية تكرار غير ضرورية تحصل في عملية حساب القيم الأعلى ارتباطية، وذلك عندما تتساوى احتمالية عنصرين، حيث تقوم خوارزمية برزم في هذه الحالة باختيار احد هذين العنصرين عشوائيا وتضيفه الى القاعدة العامة، ولكن تكمن المشكلة حين تقوم الخوارزمية في الخطوة التالية بإعادة عملية حساب الاحتمالية لجميع العناصر وتصفيتها بالكامل لتعود وتأخذ نفس العنصر الذي تساوى مع سابقه في الدورة السابقة، وهذا الاختيار ثابت وتم اثباته بالمشاهدة والمعاينة، ولكن المشكلة أن برزم تقوم بعملية التصفية والحساب لكامل العناصر مرة أخرى، هذه المشكلة تسبب تكرار غير ضروري وبالتالي تؤثر على وقت تنفيذ الخوارزمية من دون أي فائدة تذكر.

هذه الرسالة تقترح تحسين على خوارزمية برزم وذلك بإزالة هذا التكرار الحاصل في عملية توليد القواعد، سيكون ذلك من خلال اخذ العنصرين اللذين يظهران بنفس الاحتمالية ووضعهما بالقاعدة العامة في خطوة واحدة وبالتالي يتم اختصار عملية حساب الارتباطية والفلترة الإضافية الغير مجدية، وبالتالي سيقلل وقت تنفيذ الخوارزمية الإجمالي.

اضافة الى ما سبق، خوارزمية برزم لا تتعامل مع البيانات الرقمية الغير مصنفة في مجموعات او مجالات، لذلك قمنا في هذه الرسالة بإضافة عملية تصنيف مسبق الى خوارزمية برزم وذلك لمعالجة هذه البيانات، فقمنا باختيار طريقتين من طرق التصنيف المسبق وتعمدنا اختيار طريقتين مختلفتين في طريقة عملهما حتى ندرس التأثيرات الناتجة عن كل طريقة، من الجدير بالذكر قيام احد الأبحاث السابقة بمعالجة مسبقة للبيانات الرقمية، ولكن قام الباحث هناك باستخدام طرق اضافت وقت وجهد كبيرين الى خوارزمية برزم وذلك بسبب التعقيد العالي الناتج عن الحسابات الكثيرة لتلك الطرق، مما أثر سلبا على الوقت الكلي لخوارزمية برزم ، فتعمدنا في هذا البحث اختيار طرق انسب لتجنب هذه المشكلة ومحاولة تحسين استهلاك الوقت والجهد في هذه الخوارزمية.

تقليل وقت عملية توليد القواعد في خوارزمية برزم

بواسـطة
محمد ياسين حمادي

بإشـراف
د. رشيد الزبيدي

قدمت هذه الرسالة استكمالاً لـمتطلبات الحصول على درجة
الـمـاجـستير في علـم الـحـاسوب

عـمـادة البحث العلمي والدراسات العليا
جامعة فيلادلفيا

1435هـ