Event-Level Pattern Discovery
for
Large Mixed-Mode Database

by

Bin Wu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2010

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

____________________ (signature)

Bin Wu

Abstract

With the great progress of microelectronics and other relating information technologies, together with the still broadening applications of computers in a vast range of businesses and industries, large databases containing mixed-mode data are becoming quite commonplace. Today, large databases contain various modes of collected data related to different components of a complex real world system. Their use is not necessarily confined to classifications. Many of them may not have clearly-defined class labels, or even any explicit class information at all. Indeed, there are many different reasons to determine or discover all patterns, to achieve any comprehensive analysis and understanding of the information within the data spaces. In the past, data mining or pattern discovery has by and large been developed fundamentally for categorical databases. All of the classification rules have been found from pre-labeled data samples. When mixed-mode data are processed, engineers naturally work on the class-dependence relationship to discretize the real data. Where class information is lacking, there is no suitable way to discover patterns within these mixed-type databases. Consequently, most important pattern analysis jobs - such as pattern clustering, or even pattern summarization - being developed for categorical data will not be easily applied to a mixed-mode database. To break this impasse is the objective of this thesis. We have attempted to develop some pattern discovery methods for mixed-mode databases where classes or features are unavailable. Analyzing these mixed-modes of databases and providing researchers with helpful knowledge is a challenging task. Developing new ways to turn the raw data into useful knowledge is now a long-term challenge in the data mining community.

For a large mixed-mode database, how to discretize its continuous data into interval events is still a practical approach. If there are no class labels for the database, we have nohelpful correlation references to such task Actually a large relational database may contain various correlated attribute clusters. To handle these kinds of problems, we first have to partition the databases into sub-groups of attributes containing some sort of correlated relationship. This process has become known as attribute clustering, and it is an important way to reduce our search in looking for or discovering patterns Furthermore, once correlated attribute groups are obtained, from each of them, we could find the most representative attribute with the strongest interdependence with all other attributes in that cluster, and use it as a candidate like a a class label of that group. That will set up a correlation attribute to drive the discretization of the other continuous data in each attribute cluster. This thesis provides the theoretical framework, the methodology and the computational system to achieve that goal.

In validating the premises proposed in the dissertation, extensive experiments using synthetic data and UCI Expository Data of various types were performed to verify each of the fine points conceived. To demonstrate the usefulness for solving real world problems, the developed methodology is applied to two large databases from the real world: one is from meteorological surface stations, while the other is from the delay coking unit in a petrochemical refinery. The pattern discovery results of the weather stations reflect the regional and global characteristics of the correlated meteorological parameters and render a much more precise assessment of the weather monitoring system. The pattern discovery and attribute grouping experiments with the delay coking data yield the most important relationships among the sensors and controllers of the coking facilities, including the identification of the most significant control factor with global influence over the entire process, together with its interactive patterns with other factors, and with the relations discovered in the critical safety mechanism designed for a pressure-temperature-mixed processing facility, for activating emergency release response. Such findings show the usefulness and effectiveness of the proposed method in revealing subtle operation patterns for system monitoring, control and optimization.

In brief, the results of the dissertation research open the door for more precise system behavior analysis and modeling using large mixed-mode databases. It is fulfilling the vision that through pattern discovery on large mixed-mode databases, we are one step closer to meeting the challenge: "*from data to model to knowledge*" in this petabyte age.

Acknowledgements

I would like to express my deep gratitude to all those who gave me great support in completing this thesis. Here, I give great thanks to Dr. Andrew K.C. Wong for his valuable help and encouragement during my study in the University of Waterloo.

I would also like to express my deep appreciation to Dr. D. Stashuk and Dr. Yang Wang for their continuous support and help during my study in the PhD program.

Contents

Chapter 1

Introduction

In the past decade, with the development of semiconductors, microelectronics, cloud processors, magnetic storage media and other information acquisition methods, together with the continually broadening applications of computers in a wide range of businesses and industries, large databases have become quite a commonplace. The volumes of these databases have been growing from megabytes to gigabytes, and to terabytes and even to petabyte. The types of database they contain also vary: some of them could be numeric, others could be categorical and the most common ones are a mixture of both. These are referred to as “mixed-mode databases”.

Today we are facing large relational databases with mixed-mode attributes. Many of those have either no class labels, or no defined class information. They may contain different modes of correlated data, related to different attributes of a complex system. Their uses are not confined to classification. Nevertheless, there is a great need for discovering patterns among them for comprehensive analysis, interpretation and understanding the patterns or relatoins inherent in the data . Analyzing these kinds of mixed-mode databases, and thus supplying decision-makers with useful knowledge, is very challenging. Developing new measurements to transfer data into knowledge bases is now a paramount problem in data mining research community. The objective of this thesis is to develop methods for discovering patterns in mixed-mode data where class information is non-existing or unavailable.

In the past decades, data mining and pattern discovery have been developed only for categorical data. Also, inductive learning technologies have been applied widely within data mining to get classification information from a group of given data samples.

Classification rules and/or models are built, based on these pre-labeled data samples. In the early years, almost all classification tasks in data mining can only be applied to categorical data. Actually, all of these methods may not effectively handle data with continuous attributes directly.

In real practical applications, however, a large proportion of real databases may consist of not only continuous but also mixed-mode databases (continuous, discrete, ordinal as well as nominal). To make a learning system operate with these mixed types of database, these continuous attributes need first to be discretized. Furthermore, engineers have found that even if some learning systems are explicitly designed for continuous attributes, they can also maintain a relatively higher accuracy when the database can be transformed into one with appropriate discrete values. Finally, if the continuous data of the mixed-mode data attributes could be discretized appropriately, the limitations of most inductive learning algorithms may be solved by feeding the database into the current learning systems. In fact, with regards to pattern discovery and machine intelligence development today, most of the ideas available for classification in mixed-mode database require the existence of pre-labeled classes. Without that important condition, class-dependent discretization of the continuous data space will not work well, prohibtting the application of contemporary datamining methods on mixed-mode data.

Data discretization is a pre-process stage involving partitioning the value space of a continuous attribute into a finite number of intervals, and attaching a nominal value to each of them. Each interval range could then be measured as an event in the discrete data space. After discretization, we can uniformly treat both of continuous and discrete data space as events in a defined discrete event space [22].

In the mid-90s, a new class-dependent system for discretizing the value of continuous attributes was proposed [9]. It opened the way to tackling continuous data spaces for machine learning systems. It is based on a measurement of the mutual information to reflect the strength of interdependence between the continuous attributes and the class attribute. relationship.More recently, several class-dependent discretization algorithms have been developed. Most of them now can automatically determine the numbers and ranges of discretizing intervals and the discretizing interval boundaries [15], although some troubles still exist. Normally, neither the class-dependent objective functions are not effectively utilizing the class information, nor lacking an effective optimization algorithms to partition the continuous data space. .

In 2004, Liu et al [12] proposed a very effective optimization algorithm for class-dependent discretization of continuous data. It partitions the range space of a continuous random attribute into a number of ordered adjacent disjoint discrete intervals with a certain probability distribution. The expected mutual information between the class label and each of the other attributes, found by measuring the mutual information between the class and that attribute, is treated as the objective function for discretization [12]. It uses the fractional programming idea (iterative dynamic programming) to define, in a global optimum way, the expected mutual information to achieve the optimal data partitioning. Furthermore, the algorithm could efficiently partition continuous data, which was a challenging problem that had not been well solved in other ways. Bimodal and multi-modal data refer to data whose distribution measurements have respectively two or more separate and distinct peaks, each of which may correspond to a high concentration of data points in the proxmity [12].

Two important issues that have to be discussed for partitioning continuous data are the number of intervals and the ranges of the relative intervals. These two problems must be solved, either by the discretization algorithm itself or by designation by the engineers [17]. Most partition algorithms require the provision of the suitable number of data intervals by the engineers. The widths of the intervals can be defined too, through the boundaries of the discretized data intervals. A good algorithm for this purpose should usually require only a few input parameters from the operators. As a specific real-world classification problem, the available class information could provide crucial support to the discretization process. The rest of the remaining problem consists of how to partition a continuous database with continuous attributes in a mixed-mode data set [12].

1.1. The problems

For a classical machine learning system, data samples for training a typical inductive learning system are generally set up using data derived from a set of attributes. Some of these attributes characterizing an event space may be categorical, symbolic, or even with discrete values, while others may be real or continuous data attributes. Many currently-existing learning machine systems have been carefully built for processing categorical attributes values only. In the area of machine learning or data mining these inductive learning problems are usually designed to discover classificatory patterns, or just rules, based on a set of data samples [25]. Classification rulers and/or patterns are created for those pre-labeled data samples with certain prior information set up by domain experts in those areas. Thus, many traditional classification algorithms in inductive learning use carefully designed categorical data spaces. They actually cannot effectively handle continuous attributes directly. To apply inductive learning systems with these kinds of mixed-mode data space, the continuous variables must first be discretized. However, for a mixed-mode database there is still nogood solution to the unsupervised learning task, or to clustering Very large mixed-mode databases have even greater challenges [17].

The general objective of this dissertation is to meet this challenge. It attempts to solve the most fundamental problems, first of partitioning very large mixed-mode databases into smaller coherent ones, and then discretizing any continuous data without relying on explicit class labels. Once this is solved, we could extend the pattern discovery, pattern clustering, summarization, and visualization tasks to very large relational mixed-mode databases.

1.2 The Motivation

We are entering into a petabyte era , with massive distributive databases acquired from various data sources in the real world. There is a great need to get comprehensive information based on them for even better understanding and insight, such that they could be well structured and applied to analysis, classification, natural interpretation, deeper understanding, effective organization, and comprehensive summarization of the mixed-mode database. The objective of this thesis is motivated by such practical needs from the real world. Since most of the data are from a diverse sources, many of them may not be have explicit class information. Then we need pattern discovery methods which may not necessarily relying on class information. Once such methods are developed, they could be applied to data clustering, pattern discovery, pattern clustering, and any other pattern post-processing jobs for large mixed-mode data sets.

Any pattern could be described as instances of the relationships n attributes in the feature space (problem domain) - the variable level patterns - or by the direct relationships or associations among variable values - the event level patterns [7]. At the variable level, a pattern could be a mathematical relationship among attributes, usually called a model; but for the event level, a pattern may be just a subset of variable values, some maybe considered as the direct description of the collections that reflect the statistical relationship of the events in the database [9]. The proposed research here will focus on event-level patterns only, and the formal definition of pattern will be discussed in a later section. For now, an event can be just treated as a pattern of a sub database of events or measurements within mixed –mode.

With databases increasing in number and sizes, greater capacity is require to collect and analyzing data on our everyday activities in business, science, society and production. Ever-larger commercial, scientific, and industrial databases have been significantly outpacing our natural abilities to interpret and digest them [4]. Facing overwhelming data growth, the existing classical methods of data processing cannot offer us useful analysis to derive important new information and helpful knowledge. The the past, pattern discovery has been used to gain classification knowledge for classification and predication application. Lately, it attempts to discover patterns to uncover the underlying principles and behaviors of systems or phenomena in the real world from data acquired in order to reason, infer and even predict the behaviors in the same sectors. The challenges are enumerated below:

1) With respect to the structures of mixed-mode data space, they are becoming more complicated than ever before. The data values could be mixed-mode, consisting of both categorical data and continuous data. At the same time, the data dimensionality could be huge. Data could have been gathered systematically over a long period, and be piled up more-or-less randomly.

2) With respect to the quality of the mixed-mode data space, undoubtedly there are many reasons and causes in the real world for data to be collected affected by many kinds of noise. Here, probabilistic approaches must be implemented in real-world databases, instead of deterministic approaches

3) With respect to applying useful patterns discovered in real production processes, a certain kind of measurement for pattern confidence and support should be implemented to render reliable data pattern analysis results and assist in the decision-making process.

4) With respect to *a priori* real domain knowledge, in most situations it is difficult or even impossible to collect adequate domain knowledge for effective decision-making. This is definitely the truth for investigations in some new application fields. Some of special domain experts, who can support some observations and measurements to set up a domain database, but will expect to get some suggestions or evidences for data analysis results for realizing and even formulating theoretical or operational ideas. Although some domain experts who are able to set up a domain databases via meticulous observations and measurements, they still desire to get in depth suggestions and evidences from the analysis results to foster and reinforce the theoretical formulation and operation practice.

All issues mentioned above represent some open challenging problems being faced, currently investigated deeply and researched carefully by the data mining community in the recent years and naturally are taken as the very essential research motivations for this dissertation.

1.3 Rationale

Since this thesis is dealing with a new problem which has not been dealt with seriously in the past, we would like to clearly state the rationale behind the research. We would like first to identify the pending problems, and state why such problems have been at a critical impasse, slowing down the development of pattern discovery and data mining in the mixed-mode data environment.

1.3.1 Problems encountered in mixed-mode database

1.3.1.1 Problem in discretizing continuous data with no class reference

Currently, most classification algorithms in machine learning can only be applied to nominal or categorical databases. They cannot effectively be applied to deal with continuous attributes directly. In order to adopt existing inductive learning systems with mixed-mode databases, the continuous variables must first be discretizedfirst. Discretization of continuous datamay enhance classification accuracy in some ways. Today, discretization of continuous variables in pattern discovery is driven by class attribute. For those databases with no class labels, there is no easy and effective way to discretize them [21]. This limitation applies to most inductive learning algorithms for both mixed-mode and continuous data. Generally, any local discretization method starts its search of the interval boundaries at a coarse and local level at the beginning, and then refines the boundaries step-by-step later, which results in locally optimal partitions [12]. On the other hand, global discretization methods could produce a good partition result over the entire continuous instance space.

In the machine learning community, supervised and unsupervised methods are two common methods. The unsupervised (class-independent) methods simply apply a prescribed scheme to cluster the continuous data without any use of the attribute class information, whereas supervised (class-dependent) methods do take into account such class information. For discretization, theoretically, because they are directed by class information, the supervised methods could automatically get the best number of intervals for each given continuous attribute for classification purposes [21].

As a static method, it carries out one discretization pass for data for each feature separately, once the maximum number of intervals has been found. This kind of static algorithm could have also been taken as a process of merging *N* adjacent intervals at the same time until a certain threshold is reached. In fact, almost all the discretization methods discussed above are static ones [26]. Finally, static discretization methods could potentially demolish the entire complex interactions among multiple attributes.

For discretization of continuous data without class labels in an attribute group, we decide to implement the OCDD (Optimal Class-Dependent Discretization) method through replacing the class label by the *mode* or the *governing attribute* (both will be defined later) of that group. Here, the mode of an attribute group is that governing attribute for which the sum of normalized redundancy with other attributes is the highest. Thus, the data discretization process is finally formulated as an optimization problem [18]. Once the mode of an attribute group is found, we take the normalized mutual information between the mode and the variable to be discretized as the objective function, and find its maximum using fractional programming (iterative dynamic programming). Unlike the majority of class-dependent discretization methods which only find the local optimum of the objective functions, OCDD finds the global optimum.

1.3.1.2 Possible existence of unknown attribute-interdependent groups

For a large database with a large number of mixed-mode attributes, it is possible that several strongly attribute-correlated groups may exist within one data space. They could finally be found if we have an attribute clustering algorithm to do the job. In classical pattern recognition and data mining procedures, clustering is an important issue. Given a relational table, any of the conventional clustering algorithms will cluster tuples into several groups, each of which is characterized by a set of attribute values based on similarity [16]. Intuitively, tuples in a cluster are more similar to each other within the same cluster than those belonging to different clusters.

It has been shown that clustering is very helpful in many data mining tasks. In the past clustering methods are mostly developed to group samples. However, a majority of the pending problems is the data set has too many attributes which might not even be correlated. To perform pattern discovery on a large mixed-mode database, this dissertation presents a new methodology to group mixed-mode attributes that are interdependent and/or correlated to each other instead. We refer to such a process as "attribute clustering". In this sense, attributes in the same cluster are more correlated to each other, whereas attributes in different clusters are less correlated. While conventional clustering subdivides a relational table into horizontal partitions (i.e., subsets of tuples), attribute clustering subdivides it into vertical partitions (i.e., subsets of attributes) [31]. Attribute clustering is able to reduce the search dimensionality of a data mining task by allowing the algorithm to search for interesting relationships from correlated attribute subsets. It helps to build pattern models within an attribute's subspace rather than on the entire attribute space. After attributes are clustered, one can select a smaller number of more representative attributes in each attribute cluster for further analysis [29]. We refer to the process of selecting representative attributes from each attribute cluster as the “attribute repooling process”.

1.3.1.3 Attribute clustering before discretization of continuous data

Following the observation in the last paragraph, this dissertation will present an attribute clustering method which is able to group mixed-mode attributes within the database automatically, based on their interdependence, so that meaningful patterns can be discovered later. The partitioning of a relation database into attribute subgroups produces a small number of attributes, within and then across the groups, to be defined for data mining tasks. After attribute clustering, the search dimensionality of each datasetfor a data mining algorithm is reduced significantly [35]. The reduction of search dimensionality is especially important for data mining in very large mixed-mode databases, particularly in databases consisting of a huge number of attributes and a small number of samples. The situation could become even worse when the number of attributes overwhelms the number of tuples. In such cases, the patterns discovered that are actually random becomes rather higher than the usual situation. It is for the abovementioned reasons that attribute grouping is an important pre-processing stage for many data mining algorithms, to ensure effectiveness when applied to a very large database [32].

This dissertation has defined the problem of attribute clustering and introduces a new method for solving it. The proposed method will cluster all of the interdependent attributes into small clusters through optimizing a criterion function, taken from an information measure that directly reflects the interdependence between attributes [34]. By applying this algorithm to a mixed-mode database, all of the meaningful clusters of attributes within the mixed-mode database will be discovered. The grouping of attributes based on attribute interdependence relationship within a group will help directly to capture different aspects of relation patterns within each group [36].

Another important process in extracting representative attributes across a large mixed-mode attribute group is known as repooling [12]. After a large attribute group has been clustered into smaller correlated groups in the preprocessing stage, more representative governing attributes (based on their multiple statistical dependence with other attributes) in each group could be pooled together to form a new group, and the new group will therefore contain more representative information across the entire mixed-mode data space, as it is not biased towards a few governing attributes.

1.3.1.4 The necessity of identifying a governing attribute in each group to drive discretization

The rationale behind identifying a governing attribute is to find a representative attribute in a subgroup of the attributes, based on the mutual information calculation among the attributes. When the mode of an associated attribute group is identifiedfirst, it could be implemented to drive the data discretizationdiscretization of the continuous attributes in the subgroup, which would be similar to the use of the class label attribute to drive data discretization in supervised learning situations. An alternative candidate to drive the data discretization procedure is that attribute which, when assumed to be the class label, gives the highest classification rate on its categorical or discretized outcomes. We refer to the latter candidate as the "intrinsic class attribute" and the role will be evaluated in the later experiments. Both the mode and the intrinsic class attribute could be considered as the representative or the governing attribute. Both provide a good representation for that attribute group.

1.4 Special Objectives

Here, we shall outline the specific objectives of this dissertation.

1) To partition a large database into sub-databases containing attributes with greater interdependence with each other.

For a very large mixed-mode database, different subgroups of the attributes may be governed by different underlying factors. Each of the cohesive attribute groups with the mixed-mode databases could represent a certain aspect of the real world system. Whether the data discretization process of the continuous data is driven by the mode, by some implicit class attributes or by governing attributes, the mixed-mode database must first be partitioned into some coherent subgroups with strong intra-group interdependency measurements. ThusThus attribute clustering based on the mixed-mode database must first be calculated, for repooling to attribute subgroups with optimal cluster configuration.

2) To discretize continuous variables in each attribute cluster

Once the mixed-mode database is partitioned into coherent attribute subgroups (clusters), the data discretization of continuous variables in each subgroup will be processed based on the concept of mode-driven discretization. The mode of an attribute group is formally defined as that attribute which has the highest sum of interdependence value with others in the same group; it could be considered the governing attribute within the group. ThusThus the class-dependent discretization algorithm could be applied to achieve the task, if the mode is considered to be the only "governing attribute" or the "implicit class attribute".

3) To apply pattern discovery on mixed-mode Databases

After the continuous data discretization is done within each sub-database, those data can be treated as databases containing only categorical data. Any algorithm for pattern discovery can then be introduced to each sub-database, or even to the entire database platform after they are linked or joined together. The patterns discovered here may be in the general form of a subset of categorical data, interval data or even a combination of categorical and interval data. Converting mixed-mode database into an events space, the pattern algorithm is also able to process missing, noisy, outlying, and/or distorted data, or even incomplete data, more effectively.

1.5 Research Outline

In this section, we would like to outline the research carried out in furtherance of this dissertation.

1.5.1 Development of interdependence measures at different phases

For analyzing mixed-mode data effectively, it will be necessary to calculate the interdependence value between mixed-mode attributes within the database. In order to set up a unified framework for this purpose, we use the normalized mutual information measure R [22, 35] to account for the interdependence relationship between: a) two continuous attributes, b) two discrete attributes, and then c) one discrete attribute and one continuous attribute.

There are two logical process phases to using normalized mutual information in the proposed system: 1) to direct the attribute clustering of mixed-mode data space, and 2) to discretize continuous data space within each mixed-mode attribute cluster. In both phases, the discretization approach discussed above has been adopted. In the first phase, we set up as many bins as the rule of thumb allowed us, to gain accuracy in estimating the mutual information between two continuous attributes within the same subcluster. 2) In the second phase, since the final process goal of the data discretization is pattern discovery, the number of discretized intervals must be equal in the same order of the number of discrete attribute values within the attribute group. ThusWe will therefore implement the OCDD (Optimal Class-Dependent Discretization) method [37] to obtain R between a discrete-valued attribute and a continuous-valued attribute, such that the number of intervals will also be in the same order of the governing attributes or the majority of the discrete attributes. Here, we should proceed first to define the normalized mutual information between categorical data. We then will outline the data discretization process during conversion of the continuous random variables into discrete random ones for various tasks in Phase I and Phase II in a more specific manner [35]. The special algorithms developed for computing various R are listed as below:

a) **Computation module of *R* between continuous random variables**

Here accuracy and bin size will be emphasized.

b) **Computation module of *R* between a continuous attribute and a discrete attribute**

Here, an Optimum Class Dependence Discretization Algorithm (OCDD) [37] will be used to first discretize the outcome values of the continuous random variable by assuming that the discrete random variable is the class label. Once the continuous random variable is discretized successfully, we could take the pair of attributes as discrete random ones in deriving their *R* measure calculation [37].

c) **Computation module of *R* between two discrete attributes**

As for *R* between two discrete attributes, generally, we could use their corresponding alphabet size to compute *R*.

1.5.2. Identification of mode in an attribute group

In order to investigate the interdependency relationship of an attribute with all the other ones within a subgroup, the concepts of *significant multiple interdependency*, and of implementation of an algorithm to identify the *mode* which has the highest significant multiple interdependency value with all other attributes in the same group, will be introduced here [29]. Within all the *Rs* between mixed-mode attributes being calculated, both of the modes within any attribute group and the R measures between attributes could be used in the k-mode attribute clustering algorithm to cluster the attributes in a large mixed-mode database into smaller ones. The plan for developing the attribute clustering algorithm will be presented in the subsequent subsections.

1.5.3. Attribute clustering

In this dissertation, we will present a methodology to group mixed-mode attributes that will be interdependent or correlated with each other. We refer to such a process as *attribute clustering*. Within this situation, all of the attributes in one cluster should be more strongly correlated with each other, whereas the other attributes in different groups should be less strongly correlated [39]. As mentioned before, attribute clustering will be able to significantly reduce the search dimensionality of any data mining algorithm, because it is able to perform searches for interesting relationships or for construction of models in a tightly correlated subset of attributes, rather than in the entire mixed-mode attribute space. After attributes are clustered, one can select a smaller number for further analysis later.

Regarding the categorical data space, a k-mode Attribute Clustering Algorithm (ACA) has been developed. However, because of the difficulties of turning a database with mixed-mode data into one which contains only categorical data, we still have noeffective attribute clustering algorithm for a mixed-mode database [35]. One of the important challenges to this dissertation research is to develop an effective method for this purpose. We could combine some of these computation modules of R as described in the previous section into the k-mode attribute algorithm, and then build a new algorithm for clustering the attributes of the mixed-mode data. We refer to this as m-ACA which stands for Mixed-Mode Attribute Clustering Algorithm [39]. To our best knowledge, this is the first attempt which has successfully clustered attributes of mixed-mode data. Before this, there has been no published work which reports being able to achieve this task. This has left a technological gap in pattern discovery for solving such a problem with a large mixed-mode database where no class information is available for process.

1.5.4. The use of mode or "intrinsic class attribute" in an attribute group

To effectively investigate the governing attributes in an attribute subgroup, the use of the mode is important. In general, as its formal definition, the mode within a subgroup is considered to be the most representative attributes in the group. In a situation where no class information is available to us, since it is the most representative attribute in the subgroup, it becomes the only ideal candidate to drive data discretization for other continuous attributes [41]. In fact, it becomes the best way to provide insight to the subgroup, through its statistical dependence feature with other attributes in the same group.

For a problem of a classificational nature, there is another candidate to conduct data discretization. For instance, if we intend to find an attribute which most resembles a class attribute for a subgroup if it is considered to play the role of a class label, we could find a good one which will give the highest classification rate as its outcomes among all of the other attributes in the subgroup. Here, we would likely refer to such an attribute as an "intrinsic class attribute" [40].

In this dissertation, we intend to explore the characteristics and the role of these two attributes from a subgroup. It could be anticipated that the mode will result the average interdependence relationship among the entire group, and whereas the intrinsic class labels can be biased to support a certain supportive subgroup of the attributes which have the highest interdependence with this one [39]. The objective of this task is to decide which candidate will perform better as an objective one.

1.5.5. Governing attribute directed discretization

One of the major impediments to the application of pattern discovery for mixed-mode databases is that there has been no easy way (prior to this dissertation) to achieve discretization of continuous data in a database setting, when class information is absent or unavailable. This dissertation proposes a method to solve this challenging problem.

In solving discretization problems, two issues have been raised. The first one is that if a governing (or most representative) attribute really exists, we could use it to drive the discretization of all continuous attributes. The second one is regarding the state of the interdependence relationship among the attributes in the subgroup. For a very large mixed-mode database, unless a class label is given or assumed in advance, there is no reason to believe that the entire database is made up of a single correlated group, or that it is governed by a single attribute. In fact, there could be several correlated attribute groups co-existing inherently in the data set, each may share more correlated information among themselves than with others; thus it is not meaningful to use the mode of the entire data set to drive the discretization. In view of this, a more reasonable approach is to first find out whether the database could be optimally partitioned into several coherent attribute groups or not, before discretization is applied to the entire group or to each of the clustered groups. This is an important notion to be explored by this thesis.

1.5.6. Pattern discovery of mixed-mode data

After the mixed-mode database partitioning and discretization problems are solved as discussed above, a large mixed-mode database can be transformed into several smaller databases, all of which may have discrete valued data, or back into a large mixed-mode database by combining various sub-databases with discrete-valued databases. We then could apply conventional pattern discovery algorithms, or data mining methodologies which are applicable to categorical data, to this set of transformed mixed-mode data space. As a result, all of the pattern-clustering algorithms and data-grouping algorithms which have been developed for categorical data can now be applied to mixed-mode data space without the need for any class information [12]. Thus this dissertation presents a fundamental framework toward intelligent pattern discovery on large amounts of mixed-mode data without relying on prior knowledge, which in many real-world situations is not available. By discovering patterns from data sets based on such an objective measurement, the nature of the problem domain will be revealed. The patterns can then be applied to solve specific problems as being interpreted or inferred with.

1.6. Organization of the Thesis

This thesis is composed of five chapters. In the first chapter, we have already introduced the problems to be solved in this dissertation. The motivation and goal of the thesis are briefly described at the same time. We have explained the general method for pattern discovery and data analysis within mixed-mode data space, that is, first dividing the large mixed-mode data space into subspaces by attribute clustering, and then converting the mixed-mode data space into a categorical data space by mode-directed categorization, and finally, conducting pattern analysis through synthesis.

In the second chapter, a comprehensive survey of pattern discovery and data analysis for mixed-mode data space, including some existing discretization methods which are suitable for all inductive learning systems, will be introduced and reviewed; also, the advantages and disadvantages of these methods are compared and discussed in detail.

In the third chapter, we present our new pattern discovery algorithms and the framework for the mixed-mode data space, which overcomes some of the problems mentioned in the second chapter. The mathematical and theoretical foundation of this newly-developed method is presented and discussed in detail at the same time. Some information measurements, such as mutual information and interdependency redundancy rate, will take important roles in our discretization method (OCDD, Optimal Class-Dependent Discretization). In this chapter, for the best result in discretizing the continuous attributes among the mixed-data space, it has been also emphasized how to use dynamic programming methods to solve our objective functions step-by-step.

The fourth chapter gives a brief overview of the experiments in pattern discovery, based on pattern discovery for a very large mixed-mode database. Various databases, including synthetic, bi-model, and real-world databases, have been used to test this proposed method, and the performance of our method compared with other existing methods is presented. Since identical discretization preprocessing algorithms can be applied to some of the same databases we have used, it is possible for us to fairly compare the performance of the different learning systems for continuous-value learning jobs.

In the last chapter, the conclusions drawn from the experiments in the chapters above are presented. All of the advantages and disadvantages of the algorithms are discussed in detail. In addition, some possible future work and improvements in this area are pointed out and discussed, as well as additional tasks that can be done in this area.

Chapter 2

Review of Related Works

2.1 Overview

By “large mixed-mode database”, we mean a database containing data with both continuous and categorical values. In a broader coverage, the data items in the database could be a) of an ordered nature, such as a real or an integer value, or rankings which could be represented as integers, or b) of an unordered discrete nature, such as categorical items made up of symbols, terms, and/or intervals. Since it is not possible to convert unordered discrete data into continuous data, in most practical applications, continuous data unusually will be converted into interval data, so that all the data items in a mixed-mode data could be processed as discrete events, to render a uniform framework for event pattern, association, and rule discovery tasks. For historical reasons, most of the classification algorithms in the machine learning area can only be used for categorical or nominal databases. Most of these classical classification algorithms are directly able to handleneither databases including continuous values [3, 14] nor mixed-mode databases directly and effectively. However, in the real world, a large portion of data actually does contain both continuous and categorical values, or what we refer to as “mixed-mode” values. Having the current existing inductive learning systems been easily applied to these kinds of mixed-mode databases from the real world and all of the continuous values within the mixed-mode databases should be needed to be discretized first before any kind of pattern discovery analysis tasks have been conducted on them. Recently, some researchers also have found that even if some learning systems are explicitly designed and built for continuous attributes or databases, these systems still could attain a higher accuracy than unprocessed databases if continuous data are appropriately discretized. As a logical result, the limitations of most inductive learning algorithms will be overcome by discretizing all continuous attributes appropriately, before feeding those datasets into the existing learning systems [3, 6, 7, 12, 23].

Actually, any discretization could be thought of as a pre-process by which we partition the value space of a continuous attribute into a finite number of intervals, and at the same time, assign a nominal value to each of them [7]. Each interval can then be considered as a discrete event or sample, for pattern discovery at the event level. After the discretization process, all the continuous values have been converted to the discrete event space, and thus the mixed-mode database is transformed into a categorical database, much more suitable for the subsequent conventional data mining and pattern discovery tasks.

In this thesis, the researcher has described a totally new method for discretizing the values of continuous attributes within a mixed-mode database, which is entirely based on an information measurement that exactly reflects the interdependence relationship between the continuous attributes and the class attributes [6].

Traditionally, two important factors should be taken into consideration in the pre-processing phase of partitioning a continuous data space — the number of intervals, and the width of each interval. These could either be determined by the discretization algorithm itself, or provided by the system designers or operators [2]. Many existing partition algorithms for learning systems require the input of the appropriate number of discrete intervals by the system users. Alternately, the widths or boundaries of the intervals can be calculated by the boundaries of the discretized intervals during the partitioning preprocess. Naturally, the widths of intervals for discretization are determined by their boundaries. As a good algorithm, it should normally require as few inputs from the users as possible. In a specific classification task, any available class information, from the real world or domain experts, can be of crucial importance in the discretization process [3].

Some class-dependent discretization methods have been proposed [4, 29, 22, 20, 17], and most of them can automatically calculate and give the number of intervals and the interval boundaries which will be needed in the later process. Nevertheless, some challenging problems still exist with these: some class-dependent objective functions do not effectively utilize the class information within the mixed-mode database, and currently there are no effective global optimization algorithms reported for the more complex objective functions encountered in real-world situations [29].

Regarding the algorithm of OCCD, the discretization process has been viewed as the partitioning process for the data value space of a continuous random attribute into a number of ordered, adjacent, disjoint, discrete intervals with a certain probability distribution rate. The expected mutual information $I(C: A)$ between the class (C) and the attribute (A), which measures the interdependence relationship between the class and that attribute, is the definite objective function for the discretization process. Fractional programming (iterative dynamic programming) currently is adopted to calculate a global optimum value of the expected mutual information among the data in the mixed-mode database. In addition to all of the global optimization algorithms, the other important advantage of the OCCD algorithm is that OCCD can efficiently partition bimodal or even multi-modal continuous data, which is a challenging problem that has not been solved well by other partitioning methods. Usually, bimodal and multi-modal databases refer to databases whose distributions have, respectively, two or more separate and distinct peaks, each of which could correspond to a high-frequency sub-class [7].

2.2 Class-Dependent Discretization of Continuous Data

Discretization, which is an important process for transforming a continuous random attribute into an ordered discrete attribute, is a very common practice in data mining and pattern analysis tasks. Regarding the partitioning of a continuous database, two important decisions must be made before the task can be completed [8]. First, the number of discrete intervals must be determined - but the selection of the optimal number of intervals is rarely discussed in the existing literature [10]. In most situations, the system users decide to define an appropriate number of intervals at the beginning of the task. Second, the width of each interval for the discretization process must be determined. In other words, the boundaries of the intervals need to be determined before the task can proceed. Any rules or criteria for determining these interval boundaries do not usually result in a universally-applicable method. In this section, a very critical review of related works on the discretization of continuous data is presented in great detail [9].

Existing discretization schemes can be divided into two major groups. The first group, which is based on the probability density function calculation, will transform a continuous random attribute into a discrete attribute with an associated set of intervals as its discrete outputs [28]. The second group will attempt to partition or quantize [36] data into some intervals --- being similar to the first group except that the probability density function of the random attribute is unknown, and only a small set of observations on the outputs of that attribute is available. The second group is based on learning from data samples of the mixed-mode databases. This dissertation will focus on how to discretize the database from a continuous attribute which is based on observed data instances. It could be asserted that most existing algorithms, including OCCD [7], could be extended to handle the data of continuous attributes with a known probability density function distribution rate.

With most of the learning algorithms focused on nominal discrete data space, finding suitable discretization methods which can transform the data space of a continuous attribute to a finite alphabet data space will significantly improve the processing speed of the inductive learning procedure, and also will avoid over-fitting the data space. Since traditional discretization methods have been applied in clustering and classifying continuous and mixed-mode data space as early as the late 80s and even 70s, the literature on discretization topics is rich enough [42,38,41]. These are mainly divided into four groups: 1) Global versus Local, 2) Supervised versus Unsupervised, 3) Static versus Dynamic [8], and finally 4) Mulitivariate versus Univariate [2, 3].

The following sections will discuss all four groups.

Global versus Local Discretization Methods

Generally, a local method will calculate out the necessary intervals through partitioning data in one subspace or in one dimension of the instance attribute, and it will make the partition decision based on that partial information. For example, Hierarchical Maximum Entropy [6], C4.5 decision trees [29], and VQ (Vector Quantization) [17]: all of these classical discretization methods are local methods. VQ tries to divide an N-dimensional continuous data space into a discrete space, and then to represent the set of points in each interval region by the region into which the points fall [8]. The C4.5 algorithm is also a well-known example of adopting this approach, and could thus be used as a discretizer for normal discretization. This classical algorithm applies the local discretization information on the subsets of samples relative to the nodes of the decision tree during tree construction procedure. Consequently the same attribute could be discretized again by the subset of samples being available to it as the decision tree is constructed, and the final decision tree may include different partition schemes for the same attribute [29].

Since the local discretization algorithms implement many different partition methods for different portions of the sample data space, logically, one could expect them to be superior to the global methods, in producing better classification trees with generally higher accuracy. However this improved accuracy is achieved at a high cost of computational resources , as the discretization process may be repeated many times during the building of the decision tree.

Any local discretization method will start the search procedure of the interval boundaries at a coarse and raw parameter for the local level, and then gradually refine the boundaries later through a step-by-step process. Most of the local discretization methods will take advantages of heuristics to achieve an optimal solution. The final partitioning results are usually only locally optimal. Global discretization procedures, on the other hand, [16] are easily applied to the entire data space once, and thatfor all of the data space. For any of the given continuous attributes, they will be discretized first, before the mixed-mode data is fed into any machine learning algorithms. In general, global discretization methods will produce a partition result across the entire data space of a continuous attribute. Two typical examples of global discretization are the Chi-Merge [14] and *1R* (One Rule Discretizer) [12] discretization methods. The Chi-Merge method is a typical statistically-justified heuristic algorithm. It initially defines an interval to each observed value and then applies the $\chi 2$ test to determine if the adjacent two intervals should be merged together. The threshold of $\chi 2$ manages the extent and the steps of the subsequent merging process [15]. The *1R* method is a very simple classifier for discretization, which will produce a single rule known as the One-Rule. The *1R* algorithm can reach reasonably accurate results on many discretization processing tasks, through simply looking down at any attribute one at a time. This classical algorithm also attempts to reach a partitioning result such that a majority of the data space in these partitions tries to belong to only one class, that will be logical subject to a more constraint of minimal being acceptable interval widths. Holte [12] has suggested applying the *1R* method for any data space which does not contain complex relationships among the attributes within it. The *1R* method may not be a good discretization method in most situations, because its objective function is too simple to represent adequate relationships within the data space, and may well miss some important relationship when applied to real problems with more complex attribute interaction space[7]. There are many other global algorithms such as equal-width, equal-frequency, et cetera. Our discretization method is also a global one.

Supervised versus Unsupervised Discretization Methods

Supervised and unsupervised discretization methods are two very popular algorithms for discretization in the pattern discovery community. Unsupervised (also known as class-blind, or class-independent) methods will simply partition the continuous data space without any use of the attribute-class information in the data space, while supervised (also known as class-aware or class-dependent) methods will take advantage of the class information in the data space [9]. Theoretically, by using the class information within data space, the supervised methods should automatically get the optimal number of the discrete intervals for a single continuous attribute, achieving the best classification rate.

Unsupervised Methods

The simplest and most popularly-used unsupervised discretization method is to split the entire range of a continuous variable into equal-frequency [6] intervals. That can be described in the following way: given *m* instances and a user-defined number of intervals *k*, the equal frequency method here will calculate the values of a continuous attribute into the *k* bins (intervals). Each bin will contain ideally *m/k* attribute values [6].

The equal-width methods instead divide the range of a continuous variable into *k* equal-width intervals. The range is bounded by the minimum and maximum observed attribute values. The obvious weakness of the above procedures is that a large amount of important information could be lost after the discretization, where the values of a continuous variable are not distributed evenly.

To reduce the amount of lost data, a better method, based on the concept of maximum marginal entropy, has been proposed [44]. This will partition the data samples from a continuous attribute by implementing a good criterion that will maximize Shannon's information entropy, and thus try to minimize the loss of data information [23, 44]. The best number of intervals is determined by implementing a rule of thumb based on the fact that more intervals generally will lead to less information loss than fewer would. However, the method does rely on the estimation of probability distribution of the database being fed into, which is affected naturally by the sample volume or size. Furthermore, the first upper bound of the number of intervals should be limited by the second-order statistics as requiring probability estimation. Since the entire procedure of finding global maximum entropy is a highly time-consuming task, a heuristic approximation method has been developed to discretize continuous attributes, for object recognition as well as for clustering applications [44].

The algorithm of *K*-means is also another unsupervised method [36]. This algorithm will put each data point into *k* intervals according to its distance to each interval center point. This is a recursive process which finally achieves local optimization [36]. A common problem of this method is that is difficult to define which number of intervals would lead to the best decision for a specific attribute. Generally, the best or start number of the intervals must be determined by domain knowledge or by experts. In real practice, some kind of heuristics has to be employed to find the number of intervals [36].

Given that unsupervised methods will not utilize class information in calculating interval boundaries for discretization, it is more likely that some important information for classification will be lost, and as a result, values that are strongly associated with different classes might be wrongly assignedto the same interval. This could let an effective classification be much more difficult. The important advantages of these methods are that it will likely be applied to all kinds of real applications, and be put into any existing mining systems, not just restricting to classification only [33]. As a logical result, this will be taken as a significant drawback for supervised learning tasks. In addition, none of these methods have addressed the issue of the determination of the best number of intervals adequately. Too large a number of intervals is not always a good choice, because the performance of many inductive learners will deteriorate dramatically with large numbers of discrete intervals [36]. The reason for discretization is to reduce the number of possible values of an attribute, while still retaining original information from the data space as much as possible.

Despite their limitations in some situations, they are both reasonably effective, with certain specific or restrictive conditions, for most inductive leaning tasks [33].

Supervised Methods

Supervised methods utilize the class information, which in the end will place them in a leading position ahead of the existing discretization algorithms. These methods can get better partition results when compared to their unsupervised counterparts using entropy maximization method [5]. The typical methods in this group include CADD (Class-Attribute Dependent Discretization)[5], Zeta [11], Lambda [23], the Patterson-Niblett algorithm [26], Chi-Merge [21,35], Chi[13], CAIM(Class-Attribute Independent Maximum) [18], IR[47] and OCDD[24], among others. To be a good supervised discretization algorithm, it should be able to define a minimal number of discrete intervals, while retaining the interdependency relationship between attributes and the class labels as much as possible [5].

CADD will discretize a continuous data space by heuristically maximizing the interdependence relationship between the class attributes and the continuous-valued attributes [5]. The mutual information relationship between the class and the attribute which will maximally capture the interdependence is the objective function to be maximized here. Theoretically, this kind of objective function can cover the information relationship well. However it is only a heuristic search method here, which cannot guarantee an optimal solution for any situation [5].

Lambda as a supervised method is widely applied to measure association strength level between nominal attributes or variables [23]. The association strength here will be indicated by a proportionate reduction in prediction error value, that can be collected by using one attribute to predict another one by using a modal value prediction strategy among all of the applications. Lambda is an ineffective algorithm in some situations, where the dependency measurements between two attributes are not good enough to generate different modal predictions that will result in Lambda equal to zero [23]. To overcome the limitation of Lambda, Zeta, a closely related measurement, has been proposed [11]; it measures association strength level between two discretizated attributes according to the minimization of the error rate. Here, each interval value of the independent variable would predict a different interval value of the dependent variable [11]. The important and basic difference between the Lambda algorithm and the Zeta algorithm is that the latter one is not based on a modal-value-based prediction strategy, but rather on the prior assumption that each value of the independent attribute could predict a different value of a dependent one [12]. This algorithm may be generalized to the situation that one *k*-valued variable may be used to predict the values of another one which has at least *k* values. While the computation cost of the algorithm is reserved, its disadvantage is that it has the ability to handle only those kinds of attributes with a small number of values [12].

Another supervised discretization algorithm, CAIM, is similar to CADD in most ways [18]. The only different point between those two algorithms is that CAIM will use a different objective function, to not only capture the interdependency relationship between the class attribute and the continuous-valued attribute, but also to consider minimizing the number value of intervals at the same time [18].

On the whole, while all of these supervised discretization techniques might lead to more accurate classification results than expected, since they use the class information in their objective functions, they may not efficiently reach the global optimum for the objective functions. They have to rely on heuristic methods to attain local optima – usually with a heavy computational burden [18]. For this reason, one could expect that most unsupervised methods, though not as accurate as the superones, are considerably faster, because they will involve little heuristic search other than direct sorting of the data space-- an operation that is very common to all of the discretization methods. In the end, though, they will not achieve an optimal interdependence relationship between a class attribute and a continuous attribute [18].

Van de Merckt [6] has developed two effective algorithms to reduce or even remove the differences between supervised discretization methods and unsupervised discretization methods. The first method from Van de Merckt [6] is a typical unsupervised clustering algorithm that tries to find a method for generating the partition boundaries that will “produce the greatest contrast” by a given contrast measurement. The second method from Van de Merckt [6], also referred to as the mixed supervised and unsupervised methods, simply redefines the objective measurement that will be maximized by normalizing the contrast function based on the entropy of a partition result. Because calculating the information entropy for a candidate partition needs some class information, this method should be considered as a supervised one instead [6]. Chmielewski took a very similar approach through a cluster-based method to look for some candidate intervals as well as boundaries, and then evaluating the partition results based on an entropy-based consistency function [18]. By comparison with the other unsupervised methods, all of the supervised methods have their significant advantage in reaching a better partition result, because they do take advantage of using the class information. All of the supervised methods discussed here always try to have optimal interdependence information between class and attributes from the data space [6, 18].

Static versus Dynamic Discretization Methods

Regarding a static method [32], it will carry out a single discretization process for each feature separately once the maximum number of intervals is specified. This kind of discretization algorithms can be thought as a process to merge *N* adjacent intervals when a certain threshold value is found [32]. Almost all the discretization methods mentioned above are static ones, and as static discretization methods, these algorithms have not utilized the complex interactions among multiple attributes within a data space.

All of the dynamic methods [8] will search all possible numbers for the interval calculations, based on the information being collected from all of the features simultaneously. In other words, these kinds of dynamic discretization methods will determine the thresholds for a decision tree. Here, any of the dynamic methods above will discretize one attribute based not only on the information of this attribute, but also on its interaction with other attributes in the same one data space, that will explore high order relationships among the data in the space. Naturally, this will produce better partitioning results. Bay [25] has proposed a discretization method which discretizes one attribute by considering the effects of all other attributes in the database. Here, two attribute intervals should be merged together as one if the sampledata points fall into them having similar distributions. The advantage of Bay's algorithm is that all of the hidden complex patterns inside will not be destroyed by the initial discretization process [25].

In the domain of data mining, a dynamic discretization method generally is better than a static one. The reason is that a dynamic method itself will be of great interest to a data mining analyst, who may like to develop it to achieve better thresholds during the discretization pre-process. For this method, each routine procedure being crossed over the observed values of the data space could find a new partition result among the continuous data space that is based on the data intervals already being identified up to the point [27]. The method we used has adapted OCDD, which could be treated as a dynamic algorithm. It will search the entire data space for the best partition planning from all possible settings for each repeat routine [27].

The research tasks within the dissertation are highly motivated by the search for an ideal discretization method to transform continuous attributes to ordered discrete attributes, so that inductive learning and data mining techniques are able to deal with data with real values among any mixed-mode data space.

Mulitivariate versus Univariate Discretization Methods

The literature on discretization methods is abundant, but most of them are regarded as univariate method. Univariate methods consider each of the features independently (or only jointly with a class attribute). Generally, the interactions of the discretized attributes with other attributes are not considered in this way [18]. The algorithms mentioned in the above sections actually are all univariate ones. As a multivariate discretization [19], a single variable will be considered at the same time (sometimes in conjunction with the class attribute).

In fact, most of the existing discretization methods are the univariate algorithms defined by the meaning above. They only use the information being contained in a single attribute, or at most of a class label. Usually, they ignore the probable interactions of the examined attribute with other attributes [19]. However, as a multivariate discretization, it is going to become a main direction in data discretization for pattern analysis.

Dirkant and A. Grawal [34] have proposed another approach that attempts to avoid this significant limiting condition. They finely and carefully divided each of the attributes into *n* basic intervals, and then considered all of the possible combinations of these basic initial intervals. Their algorithm also encounters two challenging problems: long computation-time and too many discovered patterns for later analysis. The result of the combinatorial nature of the algorithm will be difficult to understand in later analysis procedures. Since it is almost $O(n^2)$ combination of intervals for each attribute, the computationalcomplexity will be high, especially when it takes the information interactions within other attributes from the same data space. The difficult problem of too-many-rules for understanding is also a logical result of the number of the attribute combinations. If an interval has the minimum support requirement, any range containing the interval will too.

Bay [42, 43] has significantly improved the algorithm developed by Dirkant and A. Grawal [23], and has also proposed a different multivariate discretization algorithm for continuous attributes. First, all continuous attributes will be partitioned into *n* basic intervals through a very simple discretization method, such as equal-width or even equal-frequency measurements [42]. Second, it begins to merge two adjacent intervals together that make a minimum combination support and then a very similar multivariate distribution being across all of the variables within the data space [43]. Based on comparison with those univariate discretization methods, we have found that there is a significant advantage through utilizing the interaction information among the attributes within the data space. It will have success by relying on finding a better measurement to determine how to merge the two adjacent intervals [43]. From the other viewpoints, its computation cost in time resources may be incomparable [56] with some of the existing univariate algorithms.

2.3 Attribute Clustering

During the early years when machine learning first started to be introduced , all of the researchers focused on a relatively small set of attributes in a database. With the size of real-world databases increasing and the attributes diversifying, supervised and unsupaervised learning as well as attribute clustering have begun to encounter challenging problems regarding the classification and predictive analysis. In supervised learning areas, most of the problems in this sector have been partly solved by feature selection. Even in unsupervised learning and feature clustering, the database partitioning process also was investigated as a partial solution to the problems [13]. Later, as data mining and pattern discovery have begun to come into play, the dimensionality questions have become a little relaxed, yet the ultimate problems of high dimensionality still prevail. Even now, almost all classical data clustering algorithms have to face the challenges regarding the nature of a large mixed-mode database with a large number of attributes [13]. Being diverse characteristics, a large scale mixed-mode database will often influence operational performance of any conventional clustering algorithms.

As we admitted that the problems from classification and clustering are still two major challenges for the large scale mixed-mode data analysis. While the classification mainly concern the assignment of the memberships to data instances from the discovered patterns, clustering works on finding more new implicit “class” features and keeping on refining existing ones [48]. To better cluster and then recognize patterns discovered in the large-scale mixed-mode data, the challenging problems dimensionality reduction mustt be solved. Usually, a large scale mixed-mode database has a vast number of attributes. Many of the classical data mining algorithms (such as association rule mining [10], [11], [16], [53], classification [12], [13], pattern discovery [58], [59], *context-sensitive fuzzy clustering* [26], and linguistic summaries [30]) have been developed and even optimized to overcome these kinds of difficulty, both with respect to the number of instances, and to handling a large number of attributes from a mixed-mode data space.

Following the general idea of the large database partitioning, this dissertation first works on how to cluster attributes into subgroups and then discretizing continuous attributes in each sub-group. A new methodology, called *attribute clustering for mixed-mode databases*, is introduced here, by sub-grouping the attributes which are more correlated with each other attributes within groups. Then all of the attributes within a sub cluster will be more correlated to each othersubgroup, while those in different sub-clusters are less correlated.. Here, an attribute clustering algorithm will help to solve the dimensionality problem by breaking down the original mixed-mode database into subgroups of lower dimensions. By further selecting representative attributes in each subgroup and pooled them together into a single one, pattern disocvry and data mining process are more more revealing and effective [11], [16], [53].

The proposed attribute clustering algorithm evolves from sample clustering. Even up-to-date, , the sampling clustering is still an important research issue in machine learning research.. Given a relational data table, a conventional clustering algorithm will group the data samples into some sub-clusters based on their similarity relationship [28]. Intuitively, data samples from a cluster will be more similar to others within the same cluster than they will be to those ones belonging to the other sub-clusters. However, clustering attributes is a more recent venture. It has been proven that attribute clustering is very useful in many real-world data mining application tasks (e.g., [23], [19], [47]).

Let us consider that a typical large mixed-mode database is represented by a *data table*, $T = \{w_{ij} \mid i = 1, \ldots, p, j = 1, \ldots, n\}$, where $w_{ij} \in \Re$ is the data value of the data sample g_i from the attribute s_j. Here each of the rows in this data table $T = \{w_{ij} \mid i = 1, \ldots, p, j = 1, \ldots, n\}$, will correspond to one specific data sample and each column will be an attribute in this table. While such a data table should be typically composed from a large number of the attributes, often the number of its samples might be relatively small. Hence, to handle a large scale mixed-mode database effectively, we should cluster the attributes and samples into smaller datasets [29], [19]. Talking about all those of the conventional attribute clustering algorithms, the attributes with similar expression patterns discovered to be identified [29] are acceptable and on the other hand, the similar data samples under a common data subspace of the specific attributes will be clustered together finally. Generally, both Euclidean distance and Pearson's correlation coefficient are usually adopted as the distance measurements for clustering tasks for continuous data[29], Since relation between attributes is reflected by their correlation, Euclidean distance generally used in clustering data samples is not an effective measure. Hence it is not effective to cluster attributes in a mixed-mode data space [28]. Then, Pearson's correlation coefficient is logically developed later. However, an essential study [25] has shown that Pearson's correlation coefficient is less robust for outliers and data in the presence of noise. This dissertation introduces a new technology (referred to as ACA) adapted from the *k*-modes Attribute Clustering Algorithm [22] for clustering attributes within a relational mixed-mode database. It adopts an effective similarity measurement between various types of attributes pairs from a mixed-mode data space. While the algorithm reported in [22] applies only to categorical data, the new ACA is able to be applied to mixed-mode data by implementing new information measures to evaluate the interdependence relationship between varius types of attribes in the mixed-mode data space. These mutual information measures will direct the grouping of the attributes into sub-clusters. While the ACA algorithm has been applied only to categorical data space in the past, the contribution of this dissertation is to extend ACA's capability to deal with a mixed-mode data space. A search of the literature reveals no indication that this challenging problem has ever been properly addressed or fully discussed before [23]. By implementing ACA algorithm on a large mixed-mode data space, the subgroups of the attributes based on their mutual correlation rates can be discovered and analyzed. Also, we can then still select a small part of the top-ranked attributes in each subcluster for later analysis tasks. These important attributes are generally referred to, in this dissertation, as the governing attributes of a specific sub-cluster. Choosing such a small number of the most promising attributes for model building and then pattern discovery [48] will greatly help to improve the processing speed, and should create more meaningful and reliable pattern results too.

To select significant attributes in a group , the *t*-value method is widely implemented [48]. We should note that the *t*-value can only be implemented on the data samples already pre-classified. If no class label information comes with the database, it cannot be applied to the following important attribute selection tasks. In this dissertation, we have introduced a *multiple interdependence measure* (SMI) [52], [16]) to select some of the attributes with the highest correlation rates with the other attributes within an attribute subgroup.

Various different algorithms for this important attribute clustering task have been proposed. These well-known algorithms include: *k*-means algorithms [49], [17]; Kohonen's *self-organizing maps* (SOM) [25]; and various hierarchical clustering methods [14], [21]. In the case of similarity measurements, both the Euclidean distance and Pearson's correlation coefficient rate have been widely adopted to cluster large numbers of attributes in a large mixed-mode data space [29].

Given two attributes A_i and A_j, $i, j \in \{1, \ldots, p\}$, $i \neq j$, the Euclidean distance between A_i and A_j is given by:

$$d_E(A_i, A_j) = \sqrt{\sum_{k=1}^{n} (w_{ik} - w_{jk})^2} \qquad (2.3.1)$$

where $w \in \Re$ is the measured expression level.

Here, d_E directly gives the measurement for the difference in the individual values of each attribute. Two attributes which might be similar by measuring Euclidean distance, may be dissimilar for this expression. Let us consider, for example, two attributes here, which have the same trend but differ only slightly from one another by terms of the scaling factor. Their Euclidean distance should be large, while they have the same trend by the overall trends of attributes being of basic interest in certain situations [29] - although Euclidean distance now may not be able to function as a good similarity measurement of attributes.

The Pearson's correlation coefficient between genes A_i and A_j is defined as below:

$$d_C(A_i, A_j) = \frac{\sum_{k=1}^{n} (w_{ik} - \overline{w}_i)(w_{jk} - \overline{w}_j)}{\sqrt{\sum_{k=1}^{n} (w_{ik} - \overline{w}_i)^2} \sqrt{\sum_{k=1}^{n} (w_{jk} - \overline{w}_j)^2}} \qquad (2.3.2)$$

here $\overline{w}_i$ and $\overline{w}_j$ are the means of w_{ik} and w_{jk}, k = 1, …, n, respectively. has been considered that each of the attributes being as a random one with n observations and has measured the similarity rates between the two relative attributes by calculating the linear relationship among the distributions of the two corresponding random attributes. A good study [25] has presented that Pearson's correlation coefficient is good enough to data noise and it could assign a higher similarity It score to a pair of dissimilar attributes within the same sub cluster.

Besides using the Euclidean distance, Pearson's correlation coefficient, most of the current attribute clustering methods are not as effective as the ACA even on continuous data such as gene expression data [Waiho's paper], not to mention on categorical and mixed-mode data. Hence, we adopt the ACA approach for our purpose. Thus we have to formulate the mutual information measures between various types of attributes in a mixed-mode databasesThe advantages of this information measure once implemented and validated, they can be used to direct attribute clustering, determining the mode for each cluster and eventually drive the discritization of continuous attributes. In comparison to the Euclidean distance and Pearson's correlation measures, our mutual information measures is ableto reflect both positive and negative correlational relationships among the attributes in a large mixed-mode database. The details of the information measurement and its significant features in large scale mixed-mode database correlational relationships will be discussed in the following chapters.

Feature selection is another important issue, valuable in further narrowing down the attribute number prior to pattern analysis tasks. A large number of these kinds of algorithms have been presented in the past (e.g., [44], [43]). To select the feature based on the attributes, the *t*-value is widely implemented within the literature [47]. Assume that there are two classes of data samples in a large mixed-mode database, the *t*-value $t(A_i)$ for attribute A_i is defined below:

$$t(A_i) = \frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2/n_1 + \sigma_2^2/n_2}} \qquad (2.3.4)$$

where the μ_r and σ_r represent the mean and the standard deviation of the attribute value of the attribute A_i for the class r, respectively. The n_r here is the number of samples in the class r as r = 1, 2. The top attributes ranked by the *t*-value then are selected [47]. If there are multiple classes, the *t*-value will be typically calculated for one class instead of all the other classes.

The only disadvantage to using the *t*-value to cluster attributes is the redundancy issue among the selected attributes [18]. To avoid this problem, methods that can solve both *attribute-class relevance* and *attribute-attribute redundancy* have been developed (e.g., [18],[62], [60],). These different methods mainly apply a certain metric to get the attribute-class relevance relationship (e.g., information gain, *symmetrical uncertainty* [61], mutual information, the *F*-test value [18], et c.) and then use the same or maybe a different metric to measure the attribute-attribute redundancy level (e.g., mutual information, Pearson's correlation coefficient, the L_1 distance [18], et c.). To define a subset of relevant instead of non-redundant attributes, we normally implement a new methodology called *redundant cover* to reduce redundant features based on a selected subgroup of the attributes, according to measurements of the attribute-class relevance level, and then the attribute-attribute redundancy value (see, e.g., [62], [60]). The best way to find a subgroup with relevant relationships instead of non-redundant attributes is the combination of the measures of both the attribute-class relevance and attribute-attribute redundancy as a single objective function, and then the grouping of the attributes that will maximize the function [18].

It has been noted that both the *t*-value, and the methods that process both the attribute-class relevance level and the attribute-attribute redundancy level, will only be adopted for the selection of the attributes from a mixed-mode space, as the data samples are pre-labeled before that. In this dissertation, a more general and helpful multiple interdependence measurement for attributes selecting is proposed to obtain one with the highest correlation with others.

2.4 Pattern and Association Discovery

A. Pattern Discovery

Pattern discovery, as one of the powerful intelligent decision support platforms, is being increasingly applied to large-scale complicated systems and domains even in mixed-mode space [23]. It hs been shown that it has thecapacity to extract useful knowledge from a large data space and present to the decision makers. It is growing gradually and becomes more important with the quick development of computer technologies with increasing capacity to collect massive amounts of valuable data for pattern analysis. Extracting relevant information and useful knowledge from large mixed-mode data spaces is still complicated by several challenging issues: the limitations of data storage formats; a lack of expert prior knowledge for real-world databases; the difficulty of visualizing the data using inefficient data mining tools. Data mining is a series of steps in the knowledge discovery process, consisting of the use of particular algorithms for producing patterns, as required by the real world. Useful information being extracted from real-world data using traditional data mining tools may be made better by the prior perception of a domain knowledge base or expert experience. One could use the classical data mining tools [32] to get supporting data to confirm or refute existing personal perception, but one also cannot be assured that there are no better-fitting explanations for the discovered patterns, or even that no important information has been missed in the entire data mining process. For a relatively complex real problem with a large data space, all traditional knowledge acquisition and data mining tools would become obviously inefficient, even helpless in some ways.

In decision-support,it is very easy to be biased by the subjectivities of the domain experts, or even by pre-assumptions used in data mining and the algorithmic procedures thereof. While most of the current approaches are trying to combine decision trees, neural network technologies, and the like, for pattern discovery and decision support, the rationale is to have a systematic solution providing decision-making procedure or predictive rules derived from the patterns inherent in the data space. Regarding most of the existing data mining systems, some of the accessory processes like pre-processing, data cleansing, filtering, attribute reduction are proposed [27] in order to remove data noise by bringing out more relevant information from the data space, and to reduce the search space and time, and thus cost, for that procedure.

All of the approaches discussed above make researchers investigate patterns and then verify the classification by domain experts, who often depend on their prior knowledge - including the parameters of the predetermined systematic classification framework. In that situation, they may be biased, and usually have to make long iterative search activities with personal examination and re-examination routine procedures. Due to the limited personal abilities to explore new patterns and knowledge, it is often difficult to set up a more objective base for decision-making. For a larger mixed-mode database with more unanticipated variations than normal ones, even the domain experts would find it difficult to reach useful results [27]. Furthermore, in the real world, three other important topics must be faced by the decision-makers, these being: 1) flexibility and versatility of the pattern discovery procedure; 2) transparency to get at supporting evidence; and finally 3) the processing cost and computation speed.

In conclusion, if the tools for pattern discovery could be easily implemented by the real world users, those tools should have the following basic characteristics:

1. Discover multiple patterns from a data space without relying on prior knowledge as supporting evidence;

2. Collaborate with flexible decision objectives and situations;

3. Provide significant discovered patterns for the following analysis;

4. Render a reconstruction framework with high speed of computation at low cost.

To satisfy these important and basic needs, a new pattern discovery approach has to be developed [22], which should be a primarily data-driven one. To discover an unbiased and statistically significant event automatically and exhaustively is now feasible. From theidiscovered patterns, classification modules for categorization and prediction can now be realized. At least one unique feature of the potential system is the ability to discover multiple significant patterns of high order at very fast speed, and then to list them according to their statistical confidence levels, so that a better understanding of the pattern and rules can be achieved [22].

Based on this theoretical and systematic framework design, a software platform has been developed along with several new feature modules including attribute clustering [17], class-dependent discretization [55]; classification and forecasting [43]. In this dissertation, the main emphasis has been on overcoming the difficulties in handling mixed-mode data in the new theoretical framework, and on demonstrating the performance of the new platform, especially when applying to large databases from real-world problems.

Those very initial research activities began in the early seventies by Wong [15] who first attempted to explore for quantitative information measurements and statistical patterns in English text [22], and then in digital image databases [24]. With the strong belief that information in bio-molecular data sequences is coded for bio-molecular structures, he has made a great effort to calculate quantitative information measurements and statistical patterns discovered in the bio-molecules database. It has been proved that statistical patterns discovered, which present the underlying biochemical and taxonomical features, can be identified and then analysed later. Following up this line of thought, information on quantitative measurements of how the data deviated from equal-probability and also independence models has been set up for English texts analysis [27] and images understanding [26]. These important discoveries finally formed the early basis of today’s pattern discovery approach, as discussed in this thesis. Pattern recognition algorithms for discrete continuous data space were well developed later for other real applications [28].

More recent research has noted that if the dimensionality of a real mixed-mode database is very large, this will make the definition of patterns discovered within the traditional pattern discovery framework much less meaningful [29]. Although various pattern discovery methods have been developed [44], they all depend on the interdependency of attributes with the consideration of attributes as the random variables.

In fact, all of the higher order pattern discovery platforms have been developed [45] only for discrete databases. Within those discovery frameworks, patterns have been defined as statistically significant associations of two or more primary events from different attributes in the analysis data space. For exploring patterns in databases in the presence of data noise, we have developed the adjusted residual analysis approach, which guarantees that the discovered patterns are not resulting from random association, with a fixed confidence level. All of the high-order patterns discovered can then be applied to support application tasks such as classification or pattern clustering. At the same time, the entire high-order pattern discovered within the continuous database was also advanced. Events here for the continuous data space are defined as Borel sets [45] and thus the pattern discovery is transferred into an optimization problem of finding the hypercells such that the frequency of data p oints if contains deviate statistically significantly from the default space-wise uniformity model. Analysis tasks, including classification and probability density estimation, will be easily performed based on the patterns discovered, as well as the significant analysis results on both artificial and real-world databases have been completed. These automatic pattern discovery algorithms become a good and helpful platform to support different types of decision-making tasks in the real world. As reported in [45]. while good solution to discover patterns and construct non-parametric probability density in continuous data space with scale invariant properties has been developed, the scalability for this approach is questionable because the hypercells for defining g high order significant statistical events is built on the genetic algorithm.

In this dissertation, pattern discovery theories and methodologies will be broadened to make a new framework for mixed-mode data space, which, in comparison with the approach proposed in [45], will have much more viable and scalable factors such as: 1) fast speed in discovery of significant patterns at an event level and 2) good interpretation and inference of patterns discovered in the first step.

B. Existing Methods in Association Discovery

Related to pattern discovery in the data mining community is Association Discovery. Ever since Agrawal et al. [49] defined association rules (can be considered as a special case of patterns in pattern discovery) and developed the Apriori algorithm [49], Apriori association has been widely applied to discover frequent event associations and rules in a data space for data interpretation, analysis, and understanding, by searching for interesting associations within event space by finding event associations. In this section, we will review event associations from the perspectives of data analysis. More general introductions, and surveys regarding the event association in more detail can be found in [49].

So far as data analysis is interesting, apriori association presents two major advantages: first, it can produce clearly interpretable results with the associations being readily expressed as English text or as a query task such as SQL, and this makes the mining results easily understood [47]; secondly, it works fairly well in unsupervised data mining in the case of no pre-information on this database. As the result, the approach of apriori association provides a very good starting point for the following exploration of the data space.

Many studies on this issue have been done to face the problems of having too many association patterns. Some researchers have suggested that additional specifications from the real world could be applied to help the selection of useful patterns. In [55], Silberschatz and Tuzhilin have argued that interesting patterns should be those ones unfamiliar to the end users. They then proposed a revised method that lets the end users specify their existing patterns (knowledge), and then explore only the unexpected patterns. Srikant et al. [57] also used item constraint conditions being specified by the end users to find interesting patterns. Basically, the item constraint conditions define the events which should appear in the patterns discovered. Klemetinen et al. [58] also implemented the general templates, to make the end users specify what patterns they never know or more like. All of these methods require that the end users clearly describe or define what kind of patterns they know or need.

Besides asking for more additional specifications from the end users, some researchers have also tried deleting uninteresting patterns based on certain criteria. Bayardo et al. [59] proposed to apply minimum improvement in confidence rate to reduce uninteresting patterns or associations, by comparing the confidence level between a pattern discovered and any of its simplifications, and those approaches that still do not satisfy the minimum requirement in the pattern improvement are removed. Toivonen et al. [63] also have developed a method to build up a subset of patterns that could exist across the entire data space. Other pruning methods, including pessimistic error rate [64], chi-square test [63], and minimum description length [67], have been proposed in the past.

Because the total number of patterns after removing may be still very large, how to group the discovered patterns is very useful. Toivonen et al. [68] group those ideas by applying a non-parametric density estimation algorithm. Liu et al. [54] also choose a special subgroup of patterns to build up a summary of the discovered associations. The rest of the patterns discovered are pooled together according to the summary result. Pattern pruning and grouping can be used together to further limit the total number of discovered patterns.

2.5 Pattern Clustering and Data Grouping

In response to the issue of having too many patterns and rules being discovered, one of the most significant development in pattern discovery and datamining in recent years is pattern clustering and data grouping [2] followed by pattern summarization [1]. However, the development of these methodology up-to-date applies only to categorical data. The importance of the research in this dissertation is to enable pattern clustering and data grouping to be applied to mixed-mode databases, a very significant advancement of data mining in solving the real world problems. Here a brief description of pattern clustering and data grouping will be given.

During a traditional procedure for pattern recongition, patterns are usually referred to as the pattern vectors. As databases from the real world are becoming more complex and diverse, however,, interesting information and patterns might be scattered in various data subspaces associated with different models. For this reason it is more reasonable to define the patterns in the realworld as statistically significant high-order associations of data items (events) instead of the pattern vectors in the entire feature space. After effectively discovering statistically significant patterns at a high order event level, in order to understand the way the discovered patterns are related and organized, it is beneficial to l group them first into pattern clusters, and then investigate the probabilistic variations ofeach cluster from the data group induced by the patterns in the cluster. This process is known as simultaneous pattern clustering and data grouping. Once all of these steps are finished, we could understand how those patterns relate locally to each other, and how pattern sub-groups are scattered within data subspaces.

Challenges to Existing Methods in Cluster Analysis

From the view point of data characterization, grouping or clustering can be implemented to discover the overall entire distribution patterns of the data space, by detecting correlational relationships among data attributes, by observing the characteristics of each cluster, or even by focusing on a particular cluster of entire clusters for subsequent analysis. As with the review of pattern associations, this section will discuss clustering and grouping from the perspective of the data space; more comprehensive reviews, with very detailed looks at pattern clustering, can be found in [76].

Although the conventional pattern clustering approach could be helpful for many real applications, it is still inadequate for data analysis such as interpretation and summary. If the overlapping pattern groups still exist in the database, partitioning of the data space again may divide some of the pattern groups, and destroy some very important inherent data structures. Some researchers have suggested replacing the crisp partition with a fuzzy partition approach [68]. Bezdek [69] has proposed a fuzzy *k*-means clustering algorithm that applies fuzzy pseudo-partitions instead of a crisp partition algorithm, while Tamura et al. [77] also have proposed a clustering method by which the users could adaptively determine the exact number of clusters. Other fuzzy approaches, including fuzzy learning vector quantization [78], self-organizing maps [76], and fuzzy adaptive resonance theory [69], have also been proposed during the past decade. As opposed to grouping data samples, the other approaches like attribute clustering [65] and co-clustering [66] methods will group attributes within the data space. Obviously this would help to get the best result in inevitably splitting overlapping attributes across groups. Both database partitioning [67] and bi-clustering ([69], [68]) will put data samples and attributes together, while most bi-clustering algorithms simultaneously cluster instances and attributes, as do some other algorithms such as two-way clustering [70],which will give the clusters on both separately and then combine again to get the results by obtaining bi-clusters.

The big challenges of dealing with a large scale database mainly come from the existence of irreverent attributes, and from high dimensionality across the data space. Classical approaches trying to deal with high-dimensional data spaces will include both feature transformation [83] and feature selection [87] as basic tools. Principle component analysis [85] and singular value decomposition [81] are two typical cases of feature transformation tool kits in which we do not delete irrelevant dimensions within the data space; as a result, the difficult problem of irrelevant attributes remains. The processing results after transformation are also hard to interpret for the real applications. Talking about the feature selection instead of having only those relevant attributes, to improve performance of those methods, all of heuristic methods like random searching [74] have often been implemented. Another helpful and useful method which should be mentioned here for clustering high dimensional data is subspace clustering [76], [79] by which we can search for clusters in different subspaces of database, and after this process, the clusters derived from different subspaces can be overlapped across one another.

Chapter 3

The Theoretical Framework for Pattern Discovery in Mixed-Mode Data

3.1 An Overview with Terminology and Definition

Let us begin with some of the conventions, terminologies and definitions before we introduce the theoretical framework for pattern discovery for a large mixed-mode data space. All of the terminologies and definitions provided in this chapter will be used within the entire dissertation.

As expressed in the literature review in the chapter above, patterns represented by the underlying statistically significant associations of events in data are more fundamental than others. The important advantage of pattern discovery is that it takes in only statistically significant associations up to a specified order and then, in principle, most of the statistical noise (independent events) is blocked from entering.. However, when the data space is very large, the number of discovered associations and rules may become enormous, which can make it difficult to have a comprehensive grasp of the associations at the event level inherent in the data space. Problems become more difficult to find solutions for, if information and significant events' associations might be scattered over various data subspaces. In this situation, we do need new approaches which can get into the data space to analyze, synthesize and organize local information, and also to zoom out, to extract, regroup, and organize scattered yet interrelated information or associations on a broader base.

As discussed in the literature review above, instead of considering patterns as entire vectors over the attribute space, we first define patterns as statistically significant high-order associations at the event level, in feature subspaces. For this reason, we should develop new methods to discover statistically significant local patterns (event level) effectively. In order to understand well how discovered patterns are organized within data space, it is important to find out how they are clustered via their probabilistic associations and variations in the data space. From the data induced by the patterns in the pattern cluater, we could know how the patterns are related, and how pattern groups are realized in the data subspaces.

Given a data set *D* that contains *N* tuples of mixed-mode data. Every sample is described by *N* attributes. Some of the attributes have been assigned discrete values from their own finite subset of discrete alphabet or outcomes, and some have been assigned continuous values between an upper bound *N* and a lower bound *M+1* [12].

Let $\boldsymbol{X} = \{ X_1 , \dots , X_N \}$ represent this attribute cluster from a mixed-mode data space. For convenience, we permute the attributes (without influencing the later analysis) such that the first *M* attributes $\{ Xi \mid 1 \leq i \leq M \}$ are discrete-valued attributes, and the remaining ones $\{Xi' \mid M+1 \leq \text{i'} \leq N \}$ are continuous-valued attributes. For each discrete valued attribute X_i, $1 \leq i \leq M$ can be a discrete random variable getting its values from its alphabet $\alpha_i = \{\alpha_i^1, \dots, \alpha_i^{m_i}\}$, and m_i is the cardinality of the alphabet of the *i*th attribute [17]. Each continuous-valued attribute will be represented by X_i, $M+1 \leq i \leq N$. Thus, all of the realization of $\boldsymbol{X}$ will be denoted by $x_k = \{ x_{1k} , \dots x_{ik} \dots x_{Mk,} x_{(M+1)k} , \dots x_{i'k} \dots x_{Nk} \}$ and where $\{ x_{ik} \mid 1 \leq i \leq M \}$ can assign any of the values in α_i and $\{ x_{i'k} \mid M+1 \leq i' \leq N\}$ can assign any of the values in $\{ M_{i'k} \leq \Re \leq N_{i'k}\}$,and here $\Re$ is a real number. Under this definition, each tuple from the data space will be a realization of $\boldsymbol{X}$ set.

During a petabyte era, it will be a natural situation that people will encounter in the applications of real world problems involving a massive amount of various mixed-mode types of data, which means more than ever before, the data we collect will come with the mixed-mode nature, that is, they are made up of a mixture style with discrete-valued (categorical, unordered, nominal) as well as continuous-valued (ordered, ordinal) data [22]. In the past, in both machine learning and pattern recognition researches, most of the databases gathered were just for classification purposes, or just for clustering by similarity groups according to a correlated factor from its attributes. If two subgroups of attributes are independent of each other, their usage in classification or in clustering will not be at all meaningful. This problem has been observed by Wong [11, 10, 7] in the late 70s. To resolve such a problem he has introduced the concept known as “database partitioning”, by which a database will be clustered into interdependent attribute groups first, and data clustering will then be applied to each attribute group which contains interdependent attributes only. In other words, it will be without any meaningful clusters of attributes, if those attributes have been found to be totally independent of one another. Such a measurement is also necessary for the attribute grouping with little or no interdependency with each other one [7]. In short, we will refer to the first partitioning step as “vertical partitioning” and to the ones which follow for each attribute clusters as the second step, “horizontal partitioning”. Currently, as databases grow larger and have been used to register unnecessary data for a simple classification problem, in the case where it contains diverse data of various types, this challenging problem now becomes more important. Later attribute clustering has been developed, for clustering attributes and then optimizing the intra-group attribute interdependency across the data space. Thus, to apply the pattern discovery approach to a large mixed-mode database, this challenging problem should be taken into serious consideration, and the final solution should be found.

Another important problem which must be faced when applying pattern discovery or data mining with mixed-mode data space, is how to discretize the continuous data in the mixed-mode data space [11]. This problem will be more deeply addressed, in more detail, in the following sections of this thesis.

3.2 The problems encountered and the solution proposed

As mentioned in the previous sections, the two major challenging problems encountered in current pattern discovery algorithms on large mixed-mode databases are: 1) the large attribute size; and 2) the discretization of the continuous data. In fact, these two key problems are inter-related in some ways.

As we have stated in the data discretization sections above, for an effective discretization approach, it necessarily implements the class-dependence concepts. With this viewpoint, a good discretization algorithm should maximize the interdependency between the interval values gotten from the discretization of the continuous attributes and the class labels given [17, 15]. By this reasoning, we could even apply a more effective global-optimal-class-dependent algorithm [67] for a class-dependent discretization of the continuous data space. Actually, in general in data mining and pattern discovery situations, the specific class labels may not be given, or may not be available, and then the concept of maximizing attribute-class dependence usually will not be easily applied for discretizing the continuous data space.

In this dissertation, we have proposed a new approach to tackle the discretization problem where class labels are not available in a database. Here we have to address two problems: 1) whether the data set contains attributes which characterize different subgroups within the attribute set; 2) whether the data set contains various attribute subsets, each of which contain subgroups characterized by their attributes [15]. Here the first problem is a sub-problem of the second problem..

First, we argue that for a data set containing interdependent relationships among its attributes (features), even though class labels are absent, there could still exist certain governing attribute(s) which may reflect such relationships, just as a class label reflects its dependence with other features [15]. If the ultimate objective of discretization is to reflect such interdependent relationships among the data, the resultant partition of a continuous attribute should have the highest dependence with the governing attribute as though it were a class label. In view of this, we could use the attribute with greatest interdependence with all other attributes in the group to drive the discretization of the continuous data in the group, just as in the case of using the class label to drive the discretization process [17].

The second problem arises in a more general setting, where we have no reason to believe that there is only one coherent group of attributes governed by a single governing attribute. There could be various coherent attribute groups which might even not be that interdependent with other groups. Thus to force all the continuous value data in the entire database to be discretized basing on the dependence on one governing attribute is not very reasonable [15]. Thus, before we proceed to discretization of the continuous data and subsequently to apply pattern discovery to the database, we might have to partition the databases vertically, maximizing the interdependence of the attributes within each group first. In view of this, attribute clustering should be first applied to the large database so as to group attributes together to form more coherent subgroups maximizing interdependence among attributes within the group. Once the database is clustered according to its attributes, we could treat each cluster as a coherent attribute group. Then we could proceed with the discretization of continuous data for each of them as stated in the solution of the first problem. We could either apply pattern discovery to each group, or to the data set, after the attribute groups are combined into one [7]. The second notion is useful for capturing some patterns across attribute clusters, even though the interdependence between attributes they span may be weak.

3.3 Interdependence between attributes

3.3.1 Use of interdependence measures at different phases

The major focus of this dissertation, different from other works in data mining and/or pattern discovery, is dealing with attributes which could take on categorical (discrete) and/or continuous values, that is, a mixed-mode space. Based on this viewpoint, the very basic elements required to find the interdependence among mixed-mode attributes, and all those analyses which will follow, need to take this issue into consideration. In order to use them under a unified framework, we use the normalized mutual information measurement [23, 35] to account for the interdependency between: a) discrete attributes; b) continuous attributes; and finally, c) discrete and continuous attributes.

There are two phases of using normalized mutual information. In the first phase we use it to direct the attribute clustering of mixed mode data. In the second phase, we use it to discretize continuous data within each mixed-mode attribute cluster. In both cases, we adopt the discretization approach on the continuous data. In the first phase, for more accurate approximation, we could use as many bins as we could as long as each cell resulted from the two dimensional bins contain a number of data points designated by a rule of thumbs (say two or three data points per cell). In the second phase, since the final goal of discretization is for discovering high order patterns in the mixed-mode data, there is a desirable guideline to confine the number of discretized values for each attribute cluster so that we could optimize the “intrinsic group interdependence” which will be defined later. In that case we have a unified way in defining mutual information though their implication for an attribute pair of different data mode at the different phases of the process could be different. Since all the processes of computing the normalized mutual information R become computing that between dsicretized data or categorical data, using the conventions outlined in Section 3.1, we proceed first to define the normailized mutual information between categorical data. We then outline the discretization process in converting the continuous random variables into discrete random variable for various tasks in Phase I and Phase II in a more specific manner

3.3.2 Normalized Mutual Information between Discrete-Valued Attributes.

Definition 3-1 The *interdependence redundancy measure* between two discrete –valued attributes, A_i and A_j, here, $i, j \in \{1, \ldots, M\}$, $i \neq j$, is defined below [56]:

$$R(A_i : A_j) = \frac{I(A_i : A_j)}{H(A_i, A_j)} \tag{3-1}$$

Where $I(A_i : A_j)$ is the *mutual information* between A_i and A_j, which is given by:

$$I(A_i : A_j) = \sum_{k=1}^{m_i} \sum_{l=1}^{m_j} \Pr(A_i = v_{ik} \wedge A_j = v_{jl}) \log \frac{\Pr(A_i = v_{ik} \wedge A_j = v_{jl})}{\Pr(A_i = v_{ik}) \Pr(A_j = v_{jl})} \tag{3-2}$$

and $H(A_i, A_j)$ is the *joint entropy* of A_i and A_j and is calculated by [57]:

$$H(A_i, A_j) = -\sum_{k=1}^{m_i} \sum_{l=1}^{m_j} \Pr(A_i = v_{ik} \wedge A_j = v_{jl}) \log \Pr(A_i = v_{ik} \wedge A_j = v_{jl}) \,. \tag{3-3}$$

$I(A_i : A_j)$ measures the average reduction in uncertainty about A_i that results from learning the value of A_j [36]. If $I(A_i : A_j) > I(A_i : A_h)$, $h \in \{1, \ldots, p\}$, $h \neq i \neq j$, the dependence of A_i on A_j is greater than the dependence of A_i on A_h [57].

As more accurately stated here, $R(A_i : A_j)$ reflects the degree of deviation from independence between A_i and A_j [57, 15]. If $R(A_i : A_j) = 1$, A_i and A_j are strictly dependent on each other. If $R(A_i : A_j) = 0$, that means statistically independent from each other. Also, if $0 < R(A_i : A_j) < 1$, that means A_i and A_j are partially dependent on each other [57]. The definition of the interdependence redundancy measurement shows that it is the independency of the composition of the two attributes A_i and A_j. This means that the number of attribute values will not affect the interdependence relationship and values between the two attributes A_i and A_j. The properties of the interdependence redundancy measurement clearly render an ideal candidate for measuring the dependence among different attributes within a same attribute cluster [57].

If two attributes within the same attribute cluster are dependent on one another, they will be more correlated with one another when compared to two independent attributes [33]. The interdependence redundancy measure between two attributes can evaluate the interdependence or correlation of those two attributes. If $R(A_i : A_j) > R(A_i : A_h)$, $h \in \{1, \ldots,$

p}, $h \neq i \neq j$, the dependence measurement between two attributes A_i and A_j is greater than the value between two attributes A_i and A_h. During an attribute clustering procedure, the *R* ($A_i : A_j$) is used to measure the interdependence between attributes A_i and A_j [57].

3.3.3 The mode of an attribute group

In order to investigate the interdependency of an attribute with all the other attributes within the same cluster, the concept of *significant multiple interdependency* will be defined below [15].

Definition 3-2 The *multiple interdependence redundancy measure* [15], [57] of an attribute A_i within an attribute group or cluster, $C = \{A_j \mid j = 1, \ldots, p\}$, is defined as:

$$MR(A_i) = \sum_{j=1}^{p} R(A_i : A_j)\,, \tag{3-4}$$

where $R(A_i : A_j)$ is the interdependence redundancy measure between two attributes of A_i and A_j.

Based on the definition of $MR(A_i)$, we define the concept of the "***mode***" which is a specific attribute with the highest multiple interdependence redundancy within an attribute cluster or group [15].

Definition 3-3 The ***mode*** [57] of an attribute cluster or group $C = \{ A_j \mid j = 1, \ldots, p\}$, denoted by $\eta(C)$, is an attribute, say A_i, in the cluster or group such that

$$MR(A_i) \geq MR(A_j) \text{ for all } j \in \{1, \ldots, p\}.$$

3.3.4 Computation of normalized mutual information for mixed-mode data

Currently, there are two stages for the implementation of normalized mutual information. In Phase I, it is applied to drive attribute clustering of mixed mode data. In Phase II, it is applied to direct the discretization of continuous data within each attribute sub-cluster for pattern discovery. In both stages, the discretization approach is adopted by applying the same formulas as those outlined in the Sections above [15,57]. However, there is a small difference from the computation details for those formulas. Here we will outline the computational procedures in detail for the different computations of *R*, for various processes in both Phase I and Phase II.

A. Computing *R* between continuous random variables

In phase I, to get a more accurate approximation, as many bins as possible have been used, as long as each cell results from the two dimensional bins which contain a number of data points that are designated by a rule of thumb (say at least two or three data points per cell); then it is estimated from the size of the samples to the size of the bin set [15].

Here, if *S* is the sample size and *m* is the number of bins, then the number of data points per cell could be set at *S/(m*m)* [15]. When α is defined as the parameter value in the rule of thumb (2 or 3) then [15]

$$\alpha \quad = \quad S/(m * m)$$

and therefore

$$m \quad <= \quad SQR(S/\alpha)$$

Thus, if *L* is the number of bins, then each cell in the data table will have α points. Once $m_{i'}$ is defined for all *i'* among the continuous attributes, each of them could be a discrete attribute, i.e., a random variable will have its value from its alphabet $\alpha_i = \{\alpha_i^1, ..., \alpha_i^{m_i}\}$, where m_i gets its value based on the cardinality of the alphabet, if *I, H* and *R* could be calculated between continuous attributes from equations presented above respectively [57].

B. Computing *R* between a continuous attribute and a discrete attribute

Regarding the *R* between a discrete attribute and a continuous attribute, an Optimum Class Dependence Discretization Algorithm (OCDD) [37] will be applied to first discretize the output values of the continuous random attributes, assigning the discrete random attribute as a class label for this sub-cluster. Once the continuous random attribute has been discretized, the pair of attributes could be taken as discrete random attributes in driving their *R* measure calculation [37].

C. Computing *R* between two discrete attributes

The calculation for *R* between two discrete attributes *Xi* and *Xj*, and their corresponding alphabet size m_i and $m_{j,}$ will be used for the computation of *I* and *H* from the equations mentioned above respectively [37].

3.4 Attribute Clustering of Mixed-Mode Data

During the attribute clustering procedure, a relational data table with columns for different attributes and rows for data samples is vertically divided into attribute sub-groups, which will allow a smaller number of attributes, within and/or across the subgroups, to be chosen for data or pattern analysis [15]. Through the process to cluster attributes into smaller attribute sub-groups, the search dimensionalities of a data mining algorithm for a mixed-mode are significantly reduced to a workable number. This reduction of search dimensionalities is especially critical for data mining and pattern discovery in a large mixed-mode data space, because such a database typically consists of a huge number of attributes with various data in mixed-mode types. Data mining algorithms have been typically designed and then optimized to scale to the numbers of tuples, instead of o the numbers of attributes [33]. This will become difficult to justify when the number of attributes is sufficiently larger than the number of tuples. In such a situation, the likelihood of reporting patterns that are actually irrelevant yet their occurrences are due to chances becomes rather high. It is for the aforementioned reasons that attribute grouping and selection are important preprocessing steps for many data mining algorithms to be effective when applied to large scale mixed-mode database [57]. for many data mining algorithms will be applied to large scale mixed-mode database [57]. This thesis has presented an attribute clustering method for mixed-mode data which, to our knowledge, has not been addressed before. The approach presented here will group interdependent attributes into sub clusters by optimizing a criterion function derived from an information measure that reflects the interdependence between two attributes . Although such an approach has been proposed for categorical data [44], its applications to mixed-mode data space need special preprocessing.

Another important point of the attribute clustering is that the clustering procedure has captured different aspects of association patterns discovered in various attribute sub-groups. The attribute grouping process provides a broader coverage of various attribute-interdependent sub-groups. The significant attributes being chosen from each attribute group will include information which have a broader representation of the entire database rather than information biased towards dominating group of attributes in the database[44].

Definition 3-4 *Attribute clustering* is a process which finds c disjoint clusters, $C_1, \ldots, C_c$, of correlated attributes by assigning each attribute in $\{A_1, \ldots, A_N\}$ to one of these clusters. Formally, attribute clustering will be defined as a process that $\forall\ A_i,\ i \in \{1, \ldots, N\}$, A_i will be devoted to a $C_r,\ r \in \{1, \ldots, c\}$, where $C_r \cap C_s = \emptyset$ for all $s \in \{1, \ldots, c\} - \{r\}$ [44].

To create some meaningful clusters, the attribute clustering process is made so that attributes within a sub-cluster will have higher interdependency with each other attribute, whereas the other attributes in different clusters are less correlated, or more independent, than the others [57]. Most of the conventional clustering approaches apply to cluster samples. They usually use a certain distance to measure dissimilarity between two objects like data samples whereas, for this dissertation, we will implement the clustering process to cluster interdependent attributes instead within a mixed-mode data space. To achieve such task, the k-mode approach reported in [45] has been adopted with the introduction of the new normalized interdependence information measure between a) two continuous attributes, b) discrete attribute and continuous attributes and c) two discrete attributes. With this new set of interdependence measures among the mixed-mode attributes, it is possible to complete [45] the attribute clustering algorithm for a large mixed-mode database.

To group attributes of mixed-mode data into clusters, we extend the *k*-modes algorithm developed for categorical data to mixed-mode data. By assigning an integer *k*, the *k*-mode clustering algorithm will obtain *k* clusters optimizing the intra-group attribute interdependence. To find the best choice for *k*, we use the sum of the multiple significant interdependence redundancy measure obtained for each cluster configuration by the *k*-mode algorithm. We then choose the *k* that maximizes that normalized sum of redundancy measure of the *k*-cluster configuration over all the other cluster configurations.

ACA Algorithm:[45]

the *k*-Mode Attributed Clustering Algorithm for Mixed-Mode Data

1. ***Initialization.*** Set the number of clusters as k where k is an integer greater than or equal to 2. Of the p attributes, we randomly select k attributes, each of which represents a candidate of the mode η_r for $C_{r,}$ $r \in \{1, \ldots, k\}$. Formally, let $\eta_r = A_i$, $r \in \{1, \ldots, k\}$, $i \in \{1, \ldots, p\}$, to be the mode of C_r and $\eta_r \neq \eta_s$ for all $s \in \{1, \ldots, k\} - \{r\}$.
2. ***Assignment of every attribute to one of the clusters.*** For every attribute, A_i, $i \in \{1, \ldots, p\}$, and each cluster mode, η_r, $r \in \{1, \ldots, k\}$, the interdependence redundancy measurement between A_i and η_r , $R(A_i : \eta_r)$ is calculated. Assign A_i to C_r if $R(A_i : \eta_r) \geq R(A_i : \eta_s)$ for all $s \in \{1, \ldots, k\} - \{r\}$.
3. ***Computation of mode for every attribute cluster.*** For every cluster, C_r, $r \in \{1, \ldots, k\}$, $\eta_r = A_i$ if $MR(A_i) \geq MR(A_j)$ for all $A_i, A_j \in C_r$, $i \neq j$.
4. ***Termination.*** Steps 2 and 3 are repeated until the mode η_r for each of the clusters does not change. Alternatively, ACA also terminates when the pre-specified number of iterations is reached.

It is important that the number of clusters, k, is input to the ACA algorithmWe then propose a method to choose the number k to render the best cluster configuration, that is one with k clusters [44]. To find the best value for number k, the sum of the multiple significant interdependence redundancy measure $\sum_{r=1}^{k} \sum_{A_i \in C_r} R(A_i : \eta_r)$ is used. . For every cluster configuration obtained by ACA (say with k clusters) , the overall intra-group interdependence will be evaluated by the normalized multiple interdependence redundancy measure45]. With this measure, the ACA algorithm will be run for all $k \in \{2, \ldots, p\}$.The value k that maximizes the normalized multiple interdependence redundancy measure over all the cluster configurations will be taken as one rendering the best cluster configuration [44]. When reached, the k-cluster configuration will be taken as the local optimal configuration. That is, the value of k will be selected such that [45]

$$k = \arg\max_{k \in \{2, \ldots, p\}} \sum_{r=1}^{k} \sum_{A_i \in C_r} R(A_i : \eta_r) \, . \qquad (3\text{-}5)$$

To discuss the complexity of the ACA algorithm, we set up a relational table, which include n samples such that each data sample is characterized by p attributes. The k-modes algorithm will require $O(np)$ operations to assign each attribute to a cluster (Step 2) and then it has $O(np^2)$ operations to find the mode for each cluster (Step 3) [45]. Let t represent the number of iterations, the computational complexity of the k-modes algorithm is defined by [44]:

$$\begin{aligned} O(\text{ACA}) &= O(k(np + np^2)t) \\ &= O(knp^2 t) \end{aligned}. \tag{3-6}$$

The ACA computation task can be completed in a reasonable amount of time by any modern computing machine. Furthermore, the k-modes ACA algorithm could easily be parallelized to run on a platform of clustered multi-processors, because the calculation of the interdependence redundancy measure can be performed as an independent task [45].

3.5 Mode-Driven Discretization of Continuous Data within Attribute Groups

For finding an efficient algorithm such that its partitioning result will be better for most inductive learning systems, a global optimum algorithm has been proposed in this section, which will apply the class-attribute dependency information as the criterion for final optimal discretization [23]. The discretization process could be taken as the partitioning of the value of the outputs of a continuous attribute into a number of discrete intervals, each of which can be considered as an event, and thus the attribute can be treated as an attribute with ordered discrete values. The term “global optimum” is used in the sense that the optimal partitioning of the outcome space of the continuous variable is not obtained via local perturbation but rather by applying optimization over the entire space. Actually, with only a sample of observed outputs of a continuous attribute available, the discretization process is equivalent to the reduction of the number of states of an ordered discrete random attribute by combining some of its values together [35]. Here, our approach will usea inter-dependency measure related to mutual information as a criterion function for finding the best partition intervals.

3.5.1 Class-Attribute Dependency Measurement

Before describing the discretization method, it is necessary to introduce some basic concepts for better presenting the proposed ideas. As mentioned earlier, we might know that certain information measure can be derived from data to reveal if the interdependency between attributes departs significantly from independent models. In intormation theory, mutual information plays a central role in measuring interdependence between random variables. In this dissertation, we make use of the interdependence redundancy as the objective function to drive the discretization of the continuous data into discrete interval events that maximinzing the interdependence between the discretized events and the class labels. The rationale behind is that if classification is the objective of the classificatory analysis, the partition of the continuous data optimizing the classification rate should be the best choice as observed in [23].

However, in pattern or rule discovery setting, the best classification results could be based on a subset of very strong rules or patterns. They may not necessarily represent the highest interdependence between the class labels (or the intrinsic "governing attribute" to be defined later) and the continuous attributes for the discretization of the continuous attributes. Hence, in this dissertation, we will study both options: 1) discretization driven by the mode in the sub-set and 2) decretization driven by the attribute which has the best classification rate if it is considered as the class label. Of course, of practical concerns are the size and the distribution of the intervals of the selected governing attributes. For instance, among the discrete attributes, should we choose the binary one or the multi-value one even if the former has highest *SR(i)* measure or highest classification rate [39]. In a practical setting, domain requirement might have to be considered. As for this dissertation we will confine our study to the exploration of these two criteria so as to gain some insight to the very nature of complex pattern discovery problems. As stated in the previous section, the mutual information *I (C, A)* between class and an attribute can be treated as a test of the null hypothesis of deviation from independence. Though our proposed pattern discovery do not require class labels, the proposed methodology is motivated by class-attribute discretization except that the traditional class label in the data is replaced by an attribute which have similar governing effects over other attributes resembling what a class attribute would. Hence the problem formulation still resembles supervised learning and global optimization except that the learning and optimization process is driven by a hypothetical "governing attribute" resembling the class label attributes [55]. Before then, we will set up the classification framework to formulate the discretization scheme driven by the "governing attribute".

Given a classification problem, suppose that there are *M'* samples for training, each of which has been pre-classified into one of the *K* classes c_k ($k = 1, 2, \ldots, K$). Let C_k denote the set of samples with a class label c_k. It is assumed that each one of the training samples is represented by *L* attributes A_l, $l= 1, 2, \ldots, L$ [57]. In general, it is assumed that all of the attributes A_l, $l= 1, 2, \ldots, L$ are continuous attributes. For any one of the attributes A_l, there is a range of possible values defined within the domain $\{A_l=v_{lk}|(k= 1, 2, \ldots, K)\}$, where v_{lk} could be continuous, categorical, or both. We first define that the interval $[a_l, b_l]$ is the value domain for the attribute A_l ($1 \leq l \leq L$). For the purpose of notational simplicity, we decide to use *A* to represent any attribute A_l and $[a, b]$ for its value domain (here *a* can be negative infinity and *b* can be positive infinity, in such situation, the domain could be denoted by (a, b)). Let $A^{\Psi R}$ represent a partition sample for the attribute *A* with *R* intervals where the superscript Ψ_R is a natural sequence $(e_0, e_1, \ldots, e_{R-1}, e_R)$ such that $a= e_0<e_1<\ldots<e_R =b$. All of those data sets are the boundaries of the *R* intervals respectively [57].

In general, discretization is a specific process of transforming the range of the continuous attribute *A* into a discrete partition $A^{\Psi R}$ which will have *R* intervals [60]. After the discretization process, a continuous attribute cluster can be processed as a discrete random attribute. The class label for each sample will be also processed as an output of the random attribute with class labels. We can then get a two dimensional contingency matrix [60].

As mentioned in the section above for the inductive learning systems, a training sample should consist of M data samples. Each one of the objects has been pre-classified into a specific class from a set of *K* possible classes. A specific continuous-valued attribute *A* can assign a value within a range of values. Based on the observed joint-outputs of the classes and the uniquely ordered attribute values, a 2-dimensional contingency table (Table 3-1) could be constructed.

Table 3-1 A Contingency Table between the Classes & Discretization Intervals [55]

	Interval marked by its upper bound e_r						
Class	e_1	e_2	$\cdots$	e_r	$\cdots$	e_R	Total
c_1	q_{11}	q_{12}	$\cdots$	q_{1r}	$\cdots$	q_{1R}	q_{1+}
.							.
. c_k	q_{k1}	q_{k2}	$\cdots$	q_{kr}	$\cdots$	q_{kR}	. q_{r+}
.							.
c_K	q_{K1}	q_{K2}	$\cdots$	q_{Kr}	$\cdots$	q_{KR}	q_{K+}
Total	$q+1$	$q+2$	$\cdots$	$q+r$	$\cdots$	$q+R$	M'

In the table above, element q_{kr} denotes the total number of the observed samples belonging to class $c_{k,}$ where the attribute value is in the interval between e_{r-1} and e_r. From this table, the joint probability P_{kr} for a sample belonging to class $\mathrm{c_k}$ can be calculated with attribute value in the interval demarcated by the boundary pair (e_{r-1}, e_r). Let x denote an instance of data set and x_c denote its class label and x_A the attribute value of feature A. Then the following equation can be obtained [55],

$$P_{kr} = P(x|x_C = c_k, e_{r-1} < x_A \leq e_r) = \frac{q_{kr}}{M'} \tag{3-7}$$

Here, M' presents the entirety of data samples observed.

The estimated marginal probabilities of class c_k can be found in the same way, and the estimated marginal probabilities of interval R of attribute A are respectively as follows [55],

$$P_{k+} = P(x|x_C = c_k) = \frac{q_{k+}}{M'} \tag{3-8}$$

$$P_{+r} = P(x|x_A \in e_{r-1,} e_r]) = \frac{q_{+r}}{M'} \tag{3-9}$$

where q_{k+} = Σ*R/r=1* q_{kr} and q_{+r} = Σ*R/k=1* q_{kr} . With all of these notations above, the following terms can be defined logically. (For detailed exposition, please refer to [38]).

The class-attribute (CA) mutual information between the class label C and the attribute A (with intervals as outcomes) is defined as

$$I(C:A) = -\sum_{k=1}^{K}\sum_{r=1}^{R} P_{kr} \log \frac{P_{kr}}{P_{k} + P_{+r}} \qquad (3\text{-}10)$$

I (C : A) is a measure of interdependence (or more precisely, a measurement of the expected deviation from independence) between the class label *C* and the attribute *A*. *I (C : A)* is asymptotically x_2 distribute. i.e.

$$I(C:A) \approx \frac{1}{2M'} x^{2}_{(R-1)(K-1)} \qquad (3\text{-}11)$$

with *(R-1)(K-1)* degrees of freedom. By using *I (C : A)*, we can test if C and A are statistically interdependent later via its normalized measure [38].

3.5.1.1 Class-Attribute Interdependency Redundancy Measure

Similar to the definitions in the section above, given that C and A are both considered random attributes, and the joint entropy between the class labels and the attribute variables is H(C,A), then the CA mutual information I(C:A) can be normalized by following [57] .

$$R_{CA} = \frac{I(C:A)}{H(C,A)} \qquad (3\text{-}12)$$

Here, RCA is being used as the class-attribute interdependence redundancy measurement, by which an attribute has the characteristics of normalized information rate. Clearly, we can say that $R_{CA} \geq 0$ since $I(C:A) \geq 0$ and it can easily be shown that *H(C,A) > 0*. Actually, it is well known that Shannon's entropy is bounded by the values of *0* and $+\infty$. Therefore, the equation (3.14) below is equal to 0 when *I(C:A) = 0*, which is the basic condition for total independence between *C* and *A* [55]. It can be asserted that the interdependence redundancy level between the class label and an attribute variable is equal to *0,* if the attribute does not provide any helpful relevant value for classification information.

After all of the discussions above, we can reach the conclusion that $0 \leq R_{CA} \leq 1$. If *C* and *A* are totally dependent on each other, then $R_{CA} = 1$. $R_{CA} = 0$ if *C* and *A* are totally independent from each other. More formally, the definition is set up below [57].

Definition 3-5 The *interdependence redundancy measure R* between classes C and attribute A is defined by the following equation [57],

$$R(C:A) = \frac{I(C:A)}{H(C,A)} \tag{3-13}$$

Note that both *I (C : A)* and *H (C, A)* are non-negative. Hence, the value of *R (C : A)* is non-negative as well. The values above not only depend on the number of class labels and the attribute outcomes, but also on the mutual information measure between the class label and the attribute. According to [38], *R (C : A)* represents the degree of deviation from independence between the two attributes *C* and *A*, and when *R (C : A)* = *1*, the attribute and the class label are strictly dependent on each other. When *R (C : A) = 0,* they are statistically independent from each other. When *0* < *R (C : A)* < *1*, then class label *C* and attribute *A* are partially dependent on each other. The definition for *R* presents that it is independent of the composition from both the attribute and class variable. This tells us that the number of attribute values can be reduced by keeping the interdependence relationship between the class outputs and the attribute values [38]. Thus, the discretization process could be thought of as a normal process to reduce the redundancy brought by too many possible attribute values. At the same time, the discretization process should minimize the loss of correlation between the class labels and the attribute. The properties of the interdependence redundancy measurement clearly render an ideal candidate for a class-dependent discretization criterion which can serve in the discretization method as the optimization criterion [38]. In view of this, the discretization issue could be formalized as finding the partition of attribute A such that the class-attribute interdependence redundancy measurement *R (C : A)* is maximized.

3.5.1.2 Iterative Dynamic Programming

For the real applications, we frequently need to solve an optimization problem which is a computational problem in which the objective is to find the best of all possible solutions.. More formally, we do need to find the best solution in the feasible region that has the minimum or the maximum values of the objective functions, such as the issues being described in this thesis [66]. Actually, there are many total different algorithms being applied to solve the optimization problems such as: greedy algorithm, simulated annealing and enumeration, among others. Most of these approaches cannot guarantee reaching a global optimum result, and usually achieve only a suboptimal or locally optimal solution. The reason we decided to implement iterative dynamic programming for this optimization problem is to satisfy all of the special conditions. Iterative dynamic programming is a programming technique by which an optimization problem is solved by catching sub-problem solutions, rather than recomputing them. It is a branch of nonlinear optimization for problems involving ratio functions. The problem can be described formally as follows [66],

Consider a set of solutions Z = {z}.

Let $r(z) = \frac{v(z)}{w(z)}$ where both of the *v* and *w* are the two real-world functions among a certain set *Z* and *w(z)* > 0, $\forall z \in Z$.

Then the problem is to maximize c where [66].

$$c = \max_{z \in z} r(z) \tag{3-14}$$

Let *Z** denote the choice set of optimal solutions for the problem. We first assume that the set *Z** is not empty, and the problem can be solved as a parametric problem formulated as below [57]:

Let

$$r_\lambda(z) = v(z) - \lambda_w(z) \tag{3-15}$$

then the optimization problem is expressed as a function of λ below,

$$\alpha(\lambda) = \max_{z \in Z} r_\lambda(z) \tag{3-16}$$

where $\lambda \in \Re$, and $\Re$ is a real space.

Let *Z*(λ)* denote the set of optimal solutions with the value given by *λ*, and also assume that the problem has at least one optimal solution. The problem can be solved by using Dinkelbach's algorithm described by the following steps [37]:

1. Set *k=1* at the start, then select some $z \in Z$,

2. Let $z^{(1)} = z$, and $\lambda^{(1)} = r(z^{(1)})$ respectively;

3. Solve the problem $\alpha(\lambda^{(k)})$ then select a certain $z \in Z^*(\lambda^{(k)});$

4. if $\alpha(\lambda^{(k)}) = 0$, then set $z' = z$ and $\lambda' = r(z') = v(z') / w(z')$,

5. Stop and output z' as the optimal solution;

6. Otherwise, set $z(k+1) = z$ and $\lambda^{(k+1)} = r(z^{(k+1)})$.

7. Increase $k = k + 1$ and go to step 2.

With the theoretical background above, we propose a new globally optimal algorithm for class-dependent discretization on continuous attributes.

The algorithm of *OCDD* has two important points [67]. One is that it attempts to get the maximum value of a parametric objective function by dynamic programming, and the other one is its iterative process, which takes the first component to drive towards the final globally optimum solution for the class-dependent discretization objective.

One of the most important advantages of the iterative dynamic programming approach is that it applies a process called memorization [24]. In practical operations, the problem has particular efficiency if the feasible solutions are just subsets or subsequences of the data space.

3.5.2 Global Optimal Class-Dependent Discretization

After introducing the basic class-attribute mutual information and iterative dynamic programming in detail, we can now present our globally optimal class-dependent discretization algorithm [23]. The objective function is a specific function associated with an optimization objective which can determine the quality of a problem solution. Within this proposed class-dependent discretization algorithm, we choose R(C: A) = I (C: A) / H(C, A) as the objective function, and the goal of this objective function is to maximize the mutual information between an attribute,to be discretized and the class attribute. The following iterative algorithm is adapted from [23].

ALGORITHM 3-2 [OCDD] [23]

1. Let us assume an arbitrary partition ψ of an attribute *A* first, This can be represented by a quanta matrix based upon the value of q_{k+}, q_{+r}, *M'* and p_{kr}.

2. Initialize $u= I(C: A^{\psi}) / H (C, A^{\psi})$;

3. Given *u*, calculate a new partition ψ' such that $I (C: A^{\psi'}) – uH(C, A^{\psi'})$ is maximized (This step is a key component in our algorithm);

4. Obtain a new value for *u'* by $u' = I(C: A^{\psi'}) / H(C: A^{\psi'})$;

5. Compare *u* and *u'*. If $u = u'$, then ψ' is the optimal partition.

6. Otherwise let $u = u'$ and repeat step *2, 3* and *4.*

As of now, we have introduced the algorithm which has super-linearly converges to the optimal solution. Theoretically, this approach should reach an optimal partitioning; practically, there are a number of issues or problems we need to address in real-world applications [23]. In the following section, we will discuss these problems and their impacts on the algorithm's performance.

3.5.3 Methods to Reduce the Number of Intervals

Real-world data comes with noise and outliers, and is never clean. The data noise, for example, being caused by measurement errors often produces some small intervals in the discretization process. Thish is an inherent drawback to this proposed class-dependent discretization approach [46]. Because the objective function directly relates to the dependence between classes and attributes, sometimes the total number of the partitioned intervals produced for high dependence is far too large. Thus, the proposed method alone cannot handle the high-frequency data noise, and as a results, it will output too many intervals in the discretization results.

To minimize the noise influence in the discretization process, we should consider certain data noise suppression techniques, such as binning, clustering, and regression; to handle noisy data. We first need to sort the database, and partition it into many small bins, then finally the database will be smoothed by bin means, by bin median, or by bin boundaries, and then the smoothed database can be discretized by the proposed method [46].

In this dissertation, we have proposed the following algorithm to handle the data noise, as well as the problem of having too-many-intervals. We have already discussed this discretization issue in previous chapters.

In the discretization algorithm below, we adopt an iterative dynamic programming module; it discards inferior building blocks following every single step [54]. This approach is also known as pruning and is carried out by its pruning function. While enumerating 2-way joint plans, for example, one of the nice advantages of dynamic programming is that its query optimizers, built using dynamic programming, can be extended [5].

Based on the above theory and methodology, it is known that the necessary condition for dynamic programming and the application range of this algorithm has been proved [5]. Here the the details of the procedure for solving the proposed problem is presented.

3.5.4 Smoothing the Raw Data before Partitioning

Data collected from the real world are seldom clean. They could contain various types of noise for various reasons. Usually, the discretization result could be affected by noise. To get better performance and tolerance of the algorithm, certain methods for smoothing the original data will be adopted before discretization [5]. In fact, how to implement a smoothing algorithm to decrease the impact of data noise while keeping as much information as possible is a problem to be solved, although there are many methods to choose from for data smoothing. It is a natural solution to perform a pre-processing of the originally collected database to filter out the noise before discretization [5]. The general method used to filter the data noise is to find data which satisfy some criteria conditions for noise removal. The following smoothing technique (*Algorithm Noise Filtering*) is proposed to filter the data noise.

ALGORITHM 3-3 Noisy Data Filtering [5]

1. Given two parameters (*1*,*w*) as the threshold (*t* > *1* and width *w*).

2. For the value x_i for any attribute, a segment s_i centered at x_i is defined with radius w, i.e. $s_i = \{x_{i-w},\ldots,x_{i+w}\}$.

3. Find the class label C_{max}, within the segment s_i., which occurs most frequently. Then let f_{max} denote the occurrence frequency of C_{max}. Calculate the occurrence frequency of the class label of data x_i with the segment s_i, and denote it by f_i .

4. Change the class label of x_i into C_{max} if the ratio between f_{max} and f_i is greater than the threshold t.

In practice, the smoothing result from this method for is sorted, and the sorting result is very sensitive to the value settings of threshold t and the width w. In general, the smaller the threshold givn, the more the data would be treated as data noise to be smoothed out. This might mean that some of the important class information may be lost. Also, it has been noted that the larger the width w is set, the more data will be deemed to be data noise and would be filtered out [5]. After the smoothing process, the number of data intervals could be much smaller than the resultant ones obtainable from the original database before applying the smoothing algorithm.

Theoretically, the value of width w should be related to the number of attribute classes. This means that the larger the number of attribute classes, the greater the value of w. To retain the statistical significance, the value of the width w cannot be too small for the smoothing process [5]. The value selections for the threshold t and width w are very sensitive for the smoothing algorithm ddescribed above, as they will directly affect the data smoothing and the attribute partitioning result. The above issues should be carefully considered in the algorithm as the number of class labels is generally unknown in most cases in real world problems.

While we can choose the values of these two parameters t and w based on our own experiences with good domain knowledge, we could equally well choose t by using some probabilistic technique [15]. Given the width w and the number of the classes K, if the probability that $F_{ma}x/f_i > t$ is very small (say less than five), then the class label of x_i could be regarded as noise.

For the interval merging result, the value of parameter width *w* is very subtle and cannot be set up easily. . Let us discuss the case for an attribute with too much noise and likely to be discretized into many intervals if no merging or preprocessing is carried out. Generally, the larger the *w* is, the fewer intervals we will get. But sometimes, this rule need not be too strict. If *w* is bigger (say more than 10% of the total number of attribute values), we might get more intervals than a small *w*. Conversely, for some attributes with not much noise, it is hard to determine the best *w* for them. [19]. In some specific cases, if we set the value of width *w* at a small value (say *2*) or at a large constant value (say more than *10%* of the number of attribute values), the smoothing algorithm might create more intervals than those cases without smoothing technique process. In the other cases, the value of width *w* may have little impact on the discretization processes [19].

In fact, there is also a trade-off decision in setting the value of width *w*. After the value of width *w* has been set, and the attribute has been discretized, we may have fewer intervals as a good result, but indeed we may lose certain important class-attribute information contained in the original database. Based on many experiments we have done, setting *w* to *5* is a good default value for most databases, at which better results can be expected without losing too much class-attribute information for post-processing [5].

3.6 Pattern Discovery for Mixed-Mode Data

The proposed approach for pattern discovery in this dissertation is to discretize the continuous attributes into values of discrete attributes; thus all random attributes will be treated as discrete attributes. In the following, all the definitions regarding events, event associations and patterns will be based on discrete attributes within a unified system [12].

Definition 3-5 *A* primary event *of a random attribute A_i $(1 \leq i \leq M)$ is a realization of A_i that gets a value from* α_i [12] .

The *p*th ($1 \leq p \leq m_i$) primary event of A_i is denoted by [$A_i = \alpha_i^p$], or simply x_i^p. It is assumed that two primary events, x_i^p and x_i^q, of the same attribute A_i are mutually exclusive if $p \neq q$ [12] .

Let **c** be a subset of integers {1 , … , M } containing k elements ($k \leq M$) and $\mathbf{A}^{\mathbf{c}}$ be a subset of **A** such that $\mathbf{A}^{\mathbf{c}} = \{A_i \mid i \in \mathbf{c}\}$. Then we let $\mathrm{x}^{\mathbf{c}}$ represent the realization of $\mathbf{A}^{\mathbf{c}}$ [12] .

Definition 3-6 *A* compound event *associated with the attribute set* $\mathbf{A}^{\mathbf{c}}$ *is a set of primary events instantiated by a realization* $\mathrm{x}^{\mathbf{c}}$. *The* order *of the compound event is* |**c**|. *A* sub-compound event *of* x_j^s *is a compound event* $x_j^{s'}$ $\forall$ **c'** $\subset$ **c** *and* **c'** $\neq \varnothing$ [12] .

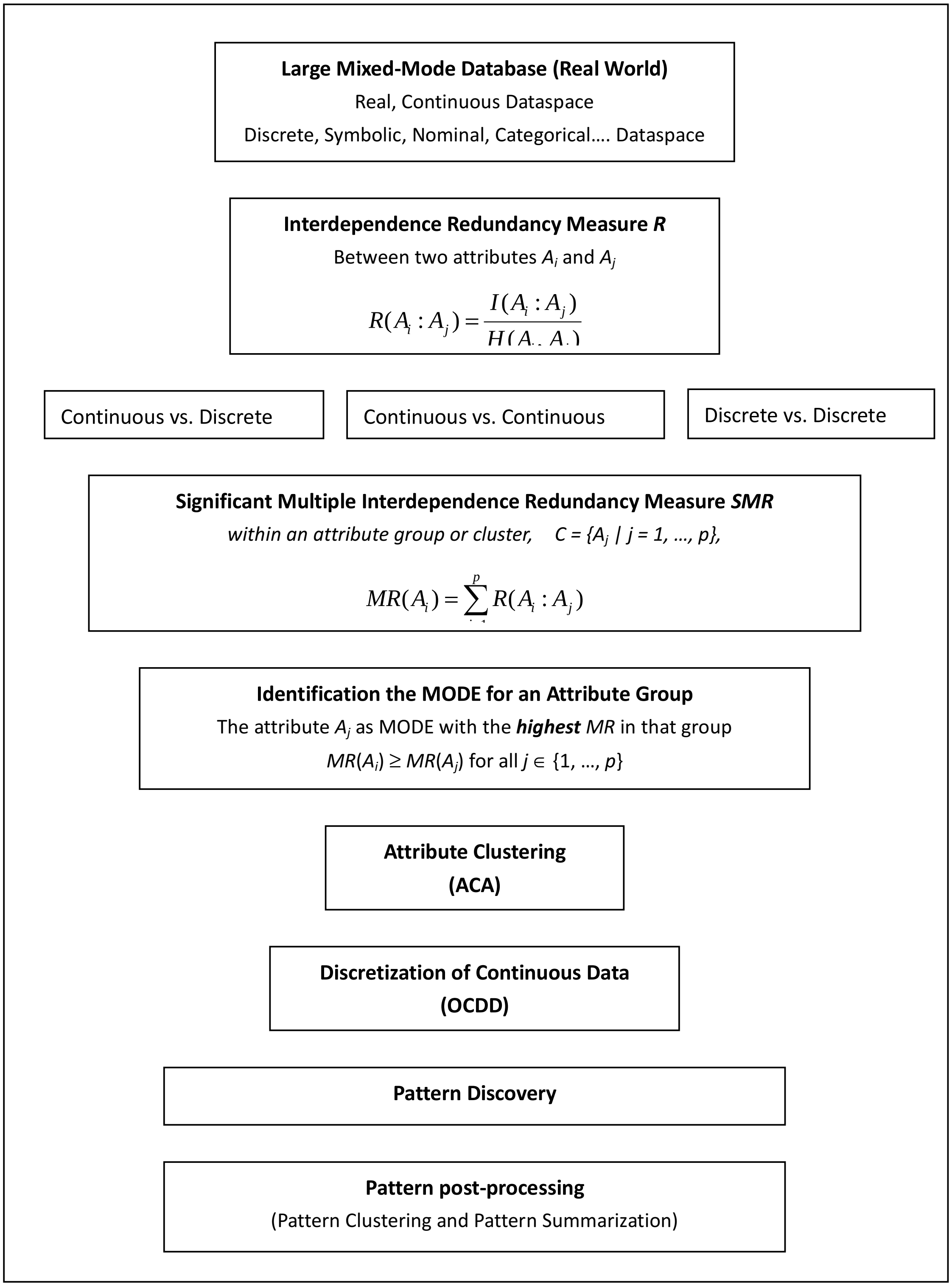
A one-compound event is a primary event in the database. A k-compound event is made up of k primary events among k distinctive attributes. Every data tuple in the database is an N-compound event [12] ..

Definition 3-7 *Let* T *be a statistical significance examination. If the occurrence of a compound event* x_j^s *is significantly different from its expectation based on a default probabilistic model, it is said that* x_j^s *is a* significant association pattern*, or simply an* association*, or a* pattern *of* order | **c** |*, and that the primary events of* x_j^s *have a* statistically significant association according to T*, or simply they are* associated [12] .

In the following context, the terms “pattern”, “significant association”, and event “association” will be used interchangeably. While pattern discovery [18] is able to discover both positive and negative patterns, our presentation and experiments will only focus on positive patterns [12]. Naturally, the occurrences of negative patterns in some databases will be discussed too, where the any of inherent patterns are definitive or deterministic sometimes.

3.7 Summary

Based on the systematic discussion in detail above, we could summarize our entire process framework for pattern discovery in a large mixed-mode data space by the following chart.


Large Mixed-Mode Database (Real World)
Real, Continuous Dataspace
Discrete, Symbolic, Nominal, Categorical…. Dataspace
Interdependence Redundancy Measure *R*
Between two attributes A_i and A_j
$R(A_i : A_j) = \frac{I(A_i : A_j)}{H(A_i, A_j)}$
Continuous vs. Discrete
Continuous vs. Continuous
Discrete vs. Discrete
Significant Multiple Interdependence Redundancy Measure *SMR*
within an attribute group or cluster, $C = \{A_j \mid j = 1, \ldots, p\}$,
$MR(A_i) = \sum^{p} R(A_i : A_j)$
Identification the MODE for an Attribute Group
The attribute A_j as MODE with the ***highest*** *MR* in that group
$MR(A_i) \geq MR(A_j)$ for all $j \in \{1, \ldots, p\}$
Attribute Clustering
(ACA)
Discretization of Continuous Data
(OCDD)
Pattern Discovery
Pattern post-processing
(Pattern Clustering and Pattern Summarization)

The general framework of Pattern Discovery for Large Mixed-Mode Database

Chapter 4

Experiments and Results

4.1. The Design of Experiments

Since this dissertation proposes a novel approach to tackle the discovery of patterns for mixed-mode data, we must design appropriate experiments to verify the premises and reveal how realistic the proposed approach when applied to various types of mixed-mode data. In this section we attempt first to design a set of experiments with selected data of various types to test our premises. Next we will apply our proposed methodologies to two large sets of real world databases which are complex, do not contain class labels but are backed by adequate domain knowledge for affirmation of the analytical results.

First we will design a comprehensive synthetic experiment with stochastically data generated to test each of the premises we proposed. We then compare and analyze the results to see whether or not our findings comply with the patterns we implanted into the synthetic data stochastically while barring out all information from the system prior to the analysis.

Next we will use various sets of UCI data with various types of data characteristics to test our proposed method. Most of the data selected are familiar in the data mining community. Since most of the data sets we choose contain class labels, they could be used as the ground truth, though not absolute, for examining the performance of the proposed method to see whether or not our method could perform its task as we anticipated and could render reasonable results even when the class labels are excluded in the analysis.

Thirdly, which is the most important task, is to apply our proposed method to two set of large real world data of mixed-mode nature. The first is a meteorology data taken from six stations located over a wide area for a relatively long span of time. The second large database is related to the processing of coke and gasoline from a delay coking plant. The database consists of a set mixed-mode data collected from in site sensors, regulators and controllers. The data was collected by the candidates with the help of domain experts. In the meantime, additional domain knowledge was acquired to see whether or not the subtle operational patterns could be discovered by the proposed system without relying on prior knowledge before the analysis.

In the design of the experiments, several questions we would like to address.

a) Are we able to optimally cluster a large mixed-mode database containing data of categorical and continuous numeric or ordered values?

b) Would the premises that certain attributes within a correlated/coherent dataset exist that reflect the characteristic of the group or could behave as one that governs the other attributes within the group like the class labels do ?

c) If there are, could the proposed method of a) mode finding and b) identifying of the attribute which plays the most representative role just like a class label be able to obtain such attributes? What are the characteristics of these governing attributes in the real world situations when the class labels are absent and how they could be related to the existing class labels? (i.e. when the class labels are put back to the data set).

d) If the governing attribute is identified within a correlated group of data, how effective is the discretization of the continuous data driven by such attribute,(i.e. optimizing the interdependence between the governing attributes and the continuous attributes).

e) After converting all the data in a mixed-mode database into discrete valued events, how effective is the pattern discovery and data mining methods when applying to a mixed-mode database?

f) How useful is the proposed method when applying to large real world mixed-mode database in revealing the inherent domain knowledge and the operation practice of the real world systems and how the discovery helps the domain experts in decision support and machine intelligence augmented operations?

g) What are the pending problems which should be solved to further enhance our method;

It is upon answering the above issues, the following experiments are designed. We hope that these experiments will shed new light to those difficult and not yet unsolved or properly solved problems.

4.2. Experiment on a Synthetic Mixed-Mode Data Set

This experiment is designed to verify the applicability of the proposed discretization method to mixed-mode data sets. It attempts to answer questions (a) to (e). It tries to demonstrate the role of the governing attributes in attribute clusters and attribute clustering and in inducing discretization of the continuous data just like the class attribute would even in the situation where the class label is absent.

Table 4.2.1 Data description of the synthetic data

Data Description				
Data Set	Attribute Characteristics	No. of Samples	No. of Attributes	No. of Classes
Synthetic Data	Mixed-Mode Data	300	20	5

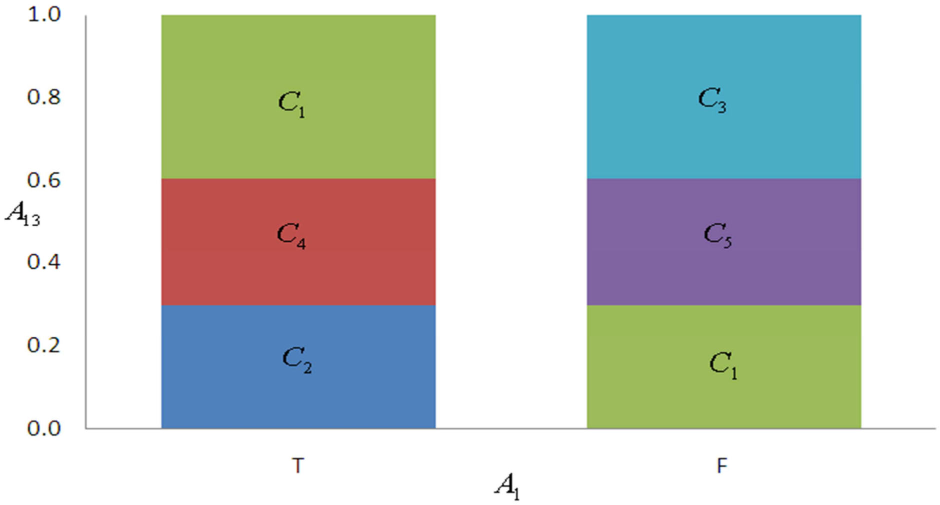


Figure 4.2.1. Imposition of intrinsic classes by adjusting the attribute values of certain attributes. In this experiment, values of attribute A_1 and A_{13} in the tuples are devised to reflect class information in the synthetic data set.

Table 4.2.1 gives a brief description of the synthetic data set with attributes made up of mixed-mode data. The synthetic data set is composed of 20 attributes in which 5 of them are discrete attributes and 15 of them are continuous attributes. Each tuple is pre-classified into one of the five classes: C_1, C_2, C_3, C_4 and C_5 by imposing the values of A_1 and A_{13} among the tuples as shown in figure 4.2.1. Let us denote the attributes as A_1, …, A_{20}. A_1 and A_2 are discrete attributes which can take on a value from alphabets {"*T*", "*F*"}. A_3, A_4 and A_5 are discrete attributes which can take on a value from alphabets {"*X*", "*Y*", "*Z*"}. A_6, …, A_{20} are continuous attributes which can take on values in $\{0 \leq \Re \leq 1\}$ where $\Re$ is a real number. As in our designed experiment, attribute values A_1 and A_{13} of each tuple are able to determine the class membership. For values of other attributes including A_2, …, A_{12} and A_{14}, …, A_{20}, they are generated randomly in the following manner:

- A_2: "*T*" if the value of $A_{13} < 0.5$; "*F*", otherwise.
- A_3: "*X*" if the value of $A_{13} < 0.5$; "*Y*" if $0.5 \leq$ the value of $A_{13} < 0.75$; "*Z*", otherwise.
- A_4: "*X*" if the value of $A_1 < 0.3$; "*Y*" if $0.3 \leq$ the value of $A_1 < 0.6$; "*Z*", otherwise.
- A_5: "*Y*" if the value of $A_1 < 0.3$; "*Z*" if $0.3 \leq$ the value of $A_1 < 0.6$; "*X*", otherwise.
- A_6-A_7: uniformly distributed within an interval between [0, 0.5] if the value of A_1 = "*T*"; uniformly distributed within an interval between (0.5, 1], otherwise.
- A_8-A_{12}: uniformly distributed within an interval between [0, 0.5] if the value of A_1 = "*F*"; uniformly distributed within an interval between (0.5, 1], otherwise.
- A_{14}-A_{17}: uniformly distributed within an interval between [0, 0.3) if the value of $A_{13} < 0.3$; uniformly distributed within an interval between [0.3, 0.6) if $0.3 \leq$ the value of $A_{13} < 0.6$; uniformly distributed within an interval between [0.6, 1], otherwise.
- A_{18}-A_{20}: uniformly distributed within an interval between [0.3, 0.6) if the value of $A_{13} < 0.3$; uniformly distributed within an interval between [0.6, 1] if $0.3 \leq$ the value of $A_{13} < 0.6$; uniformly distributed within an interval between [0, 0.3), otherwise.

Using this scheme to generate the synthetic data set, it is clear that $\boldsymbol{A_1}$ and $\boldsymbol{A_{13}}$ are two governing attributes correlating with the attribute groups of {A_4-A_{12}} and {A_2, A_3, A_{14}-A_{20}} respectively. Regardless of the class membership of each tuple, if such correlation can be revealed, one should seek the most discriminative/representative attribute of each attribute group to drive the discretization of the continuous attributes. In our experiment, we generated 300 tuples where each class contains 50 tuples in the synthetic data set. Noises are then added noises there by replacing 25 percent of the tuples with a random real number between 0 and 1 in the continuous attributes, with a random alphabet of "*T*" or "*F*" in A_1 and A_2 and, with a random alphabet of "*X*", "*Y*" or "*Z*" in A_4-A_6.

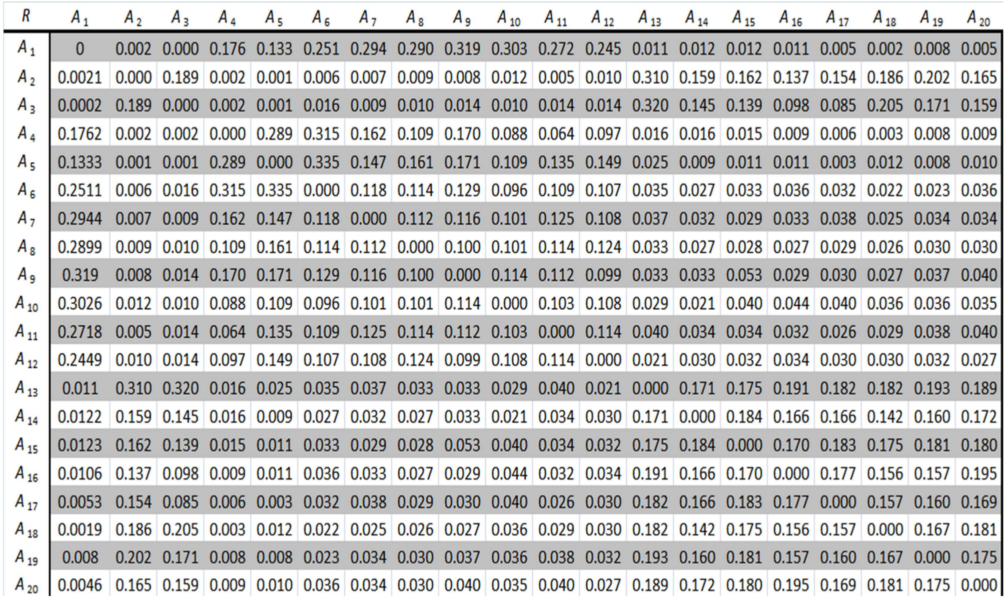
Table 4.2.1 gives a brief description of the synthetic data set with attributes made up of mixed-mode data. The synthetic data set is composed of 20 attributes in which 5 of them are discrete attributes and 15 of them are continuous attributes. Each tuple is pre-classified into one of the five classes: C_1, C_2, C_3, C_4 and C_5 by imposing the values of A_1 and A_{13} among the tuples as shown in figure 4.2.1. Let us denote the attributes as A_1, …, A_{20}. A_1 and A_2 are discrete attributes which can take on a value from alphabets {"*T*", "*F*"}. A_3, A_4 and A_5 are discrete attributes which can take on a value from alphabets {"*X*", "*Y*", "*Z*"}. A_6, …, A_{20} are continuous attributes which can take on values in $\{0 \leq \Re \leq 1\}$ where $\Re$ is a real number. As in our designed experiment, attribute values A_1 and A_{13} of each tuple are able to determine the class membership. For values of other attributes including A_2, …, A_{12} and A_{14}, …, A_{20}, they are generated randomly in the following manner:

- A_2: "*T*" if the value of $A_{13} < 0.5$; "*F*", otherwise.
- A_3: "*X*" if the value of $A_{13} < 0.5$; "*Y*" if $0.5 \leq$ the value of $A_{13} < 0.75$; "*Z*", otherwise.
- A_4: "*X*" if the value of $A_{13} < 0.3$; "*Y*" if $0.3 \leq$ the value of $A_{13} < 0.6$; "*Z*", otherwise.
- A_5: "*Y*" if the value of $A_{13} < 0.3$; "*Z*" if $0.3 \leq$ the value of $A_{13} < 0.6$; "*X*", otherwise.
- A_6-A_7: uniformly distributed within an interval between [0, 0.5] if the value of A_1 = "*T*"; uniformly distributed within an interval between (0.5, 1], otherwise.
- A_8-A_{12}: uniformly distributed within an interval between [0, 0.5] if the value of A_1 = "*F*"; uniformly distributed within an interval between (0.5, 1], otherwise.
- A_{14}-A_{17}: uniformly distributed within an interval between [0, 0.3) if the value of $A_{13} < 0.3$; uniformly distributed within an interval between [0.3, 0.6) if $0.3 \leq$ the value of $A_{13} < 0.6$; uniformly distributed within an interval between [0.6, 1], otherwise.
- A_{18}-A_{20}: uniformly distributed within an interval between [0.3, 0.6) if the value of $A_{13} < 0.3$; uniformly distributed within an interval between [0.6, 1] if $0.3 \leq$ the value of $A_{13} < 0.6$; uniformly distributed within an interval between [0, 0.3), otherwise.

Using this scheme to generate the synthetic data set, it is clear that $\boldsymbol{A_1}$ and $\boldsymbol{A_{13}}$ are two governing attributes correlating with the attribute groups of {A_6-A_{12}} and {A_2-A_5, A_{14}-A_{20}} respectively. Regardless of the class membership of each tuple, if such correlation can be revealed, one should seek the most discriminative/representative attribute of each attribute group to drive the discretization of the continuous attributes. In our experiment, we generated 300 tuples where each class contains 50 tuples in the synthetic data set. Noises are then added noises there by replacing 25 percent of the tuples with a random real number between 0 and 1 in the continuous attributes, with a random alphabet of "*T*" or "*F*" in A_1 and A_2 and, with a random alphabet of "*X*", "*Y*" or "*Z*" in A_3-A_5.

The normalized mutual information measure as defined in Table 4.2.1 between pairs of discrete attributes, pairs of continuous attributes and pairs of discrete and continuous attributes are calculated as shown in Table 4.2.2.

Table 4.2.2 Normalized Mutual Information between Mixed-Mode Attributes of the Synthetic Data



R	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}	A_{19}	A_{20}
A_1	0	0.002	0.000	0.176	0.133	0.251	0.294	0.290	0.319	0.303	0.272	0.245	0.011	0.012	0.012	0.011	0.005	0.002	0.008	0.005
A_2	0.0021	0.000	0.189	0.002	0.001	0.006	0.007	0.009	0.008	0.012	0.005	0.010	0.310	0.159	0.162	0.137	0.154	0.186	0.202	0.165
A_3	0.0002	0.189	0.000	0.002	0.001	0.016	0.009	0.010	0.014	0.010	0.014	0.014	0.320	0.145	0.139	0.098	0.085	0.205	0.171	0.159
A_4	0.1762	0.002	0.002	0.000	0.289	0.315	0.162	0.109	0.170	0.088	0.064	0.097	0.016	0.016	0.015	0.009	0.006	0.003	0.008	0.009
A_5	0.1333	0.001	0.001	0.289	0.000	0.335	0.147	0.161	0.171	0.109	0.135	0.149	0.025	0.009	0.011	0.011	0.003	0.012	0.008	0.010
A_6	0.2511	0.006	0.016	0.315	0.335	0.000	0.118	0.114	0.129	0.096	0.109	0.107	0.035	0.027	0.033	0.036	0.032	0.022	0.023	0.036
A_7	0.2944	0.007	0.009	0.162	0.147	0.118	0.000	0.112	0.116	0.101	0.125	0.108	0.037	0.032	0.029	0.033	0.038	0.025	0.034	0.034
A_8	0.2899	0.009	0.010	0.109	0.161	0.114	0.112	0.000	0.100	0.101	0.114	0.124	0.033	0.027	0.028	0.027	0.029	0.026	0.030	0.030
A_9	0.319	0.008	0.014	0.170	0.171	0.129	0.116	0.100	0.000	0.114	0.112	0.099	0.033	0.033	0.053	0.029	0.030	0.027	0.037	0.040
A_{10}	0.3026	0.012	0.010	0.088	0.109	0.096	0.101	0.101	0.114	0.000	0.103	0.108	0.029	0.021	0.040	0.044	0.040	0.036	0.036	0.035
A_{11}	0.2718	0.005	0.014	0.064	0.135	0.109	0.125	0.114	0.112	0.103	0.000	0.114	0.040	0.034	0.034	0.032	0.026	0.029	0.038	0.040
A_{12}	0.2449	0.010	0.014	0.097	0.149	0.107	0.108	0.124	0.099	0.108	0.114	0.000	0.021	0.030	0.032	0.034	0.030	0.030	0.032	0.027
A_{13}	0.011	0.310	0.320	0.016	0.025	0.035	0.037	0.033	0.033	0.029	0.040	0.021	0.000	0.171	0.175	0.191	0.182	0.182	0.193	0.189
A_{14}	0.0122	0.159	0.145	0.016	0.009	0.027	0.032	0.027	0.033	0.021	0.034	0.030	0.171	0.000	0.184	0.166	0.166	0.142	0.160	0.172
A_{15}	0.0123	0.162	0.139	0.015	0.011	0.033	0.029	0.028	0.053	0.040	0.034	0.032	0.175	0.184	0.000	0.170	0.183	0.175	0.181	0.180
A_{16}	0.0106	0.137	0.098	0.009	0.011	0.036	0.033	0.027	0.029	0.044	0.032	0.034	0.191	0.166	0.170	0.000	0.177	0.156	0.157	0.195
A_{17}	0.0053	0.154	0.085	0.006	0.003	0.032	0.038	0.029	0.030	0.040	0.026	0.030	0.182	0.166	0.183	0.177	0.000	0.157	0.160	0.169
A_{18}	0.0019	0.186	0.205	0.003	0.012	0.022	0.025	0.026	0.027	0.036	0.029	0.030	0.182	0.142	0.175	0.156	0.157	0.000	0.167	0.181
A_{19}	0.008	0.202	0.171	0.008	0.008	0.023	0.034	0.030	0.037	0.036	0.038	0.032	0.193	0.160	0.181	0.157	0.160	0.167	0.000	0.175
A_{20}	0.0046	0.165	0.159	0.009	0.010	0.036	0.034	0.030	0.040	0.035	0.040	0.027	0.189	0.172	0.180	0.195	0.169	0.181	0.175	0.000

As shown in Figure 4.2.2, the optimal attribute cluster configuration (no. of attribute clusters) obtained by ACA is two (k =2). ACA identifies two attribute clusters: {A_1, A_6, …, A_{12}} and {A_2-A_5, A_{13}, …, A_{20}}. It shows that the proposed discretization algorithm is able to correctly compute the mutual information between a pair of continuous attributes, and between a discrete attribute and a continuous attribute for ACA to reveal the correlation between the mixed-mode attributes embedded in the synthetic data set. It was found that A_1 is the mode of the first cluster whereas A_{13} is the mode of the second cluster, indicating that the attributes with the most of the intrinsic governing or classificatory characteristics are found as the mode.

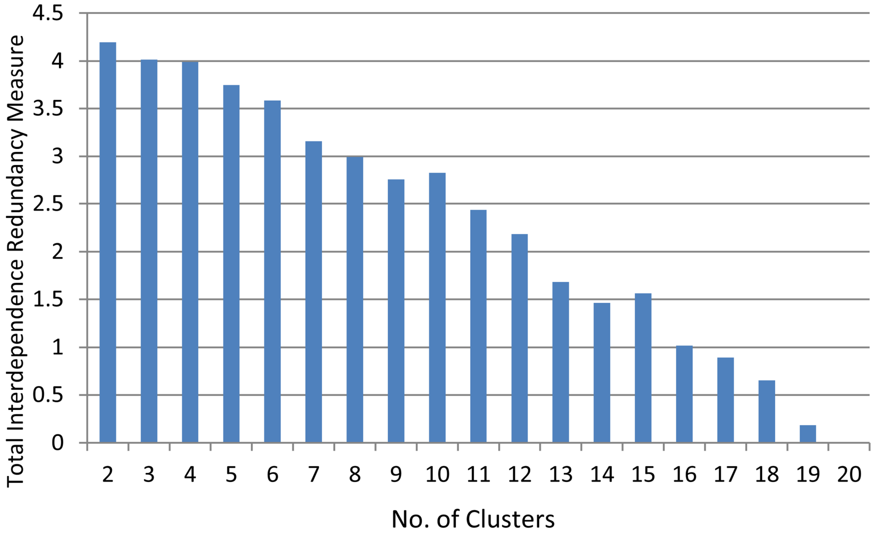


Figure 4.2.2 The Total Interdependence Redundancy Measure across the Clusters Found in the Synthetic Data Set.

To evaluate the effectiveness of the generated discretization schemes on the performance of the classification algorithm, we used the discretized synthetic data set with 25% noise to train C5.0. 30% of samples are randomly selected from the data set as the training data to build a decision tree and the rest of samples are treated as the testing data. The comparison results in Table 4.2.3 show that the proposed method reached highest classification accuracy. It is worth noting that the discretization scheme generated by the proposed method can improve classification accuracy even when the class label is excluded. As regards to the number of generated rules/nodes, the proposed method also achieves the best performance while C5.0 produced significantly more nodes when using the discretization scheme of OCDD which makes use of class label (Table 4.2.3).

Table 4.2.3. The Comparison of Discretization Schemes on Synthetic Data Set

Discretization Method	Classification Accuracy	Leaf Nodes	Non Leaf Nodes
OCDD (Discretized by Class Label)	74%	17 leaf nodes	10 non leaf nodes
Proposed Approach (Class Label Excluded)	83.67%	13 leaf nodes	10 non leaf nodes

4.3. Experiment on UCI Data Sets

4.3.1 Iris Plants Database

The objective of this experiment is to show how the proposed method is able to be applied to continuous data where the class labels are missing and how the experimental results are related to the ground truth provided by the removed class labels. This experiment attempts to answer questions: (b) to (e). Because of the transparency characteristics of pattern discovery, new light could be shed to reveal how the governing attributes are related to the correlated aspects of the attributes and also with the class labels.

The IRIS data set was created by R.A. Fisher, donated by Michael Marshall, dated July, 1988 widely used by many and is perhaps the best known database to be found in the pattern recognition literature. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two; the latter are not linearly separable from each other. The number of instances is 150 (50 in each of the three classes). It contains 4 numeric attributes:

1) sepal length in cm; 2) sepal width in cm;

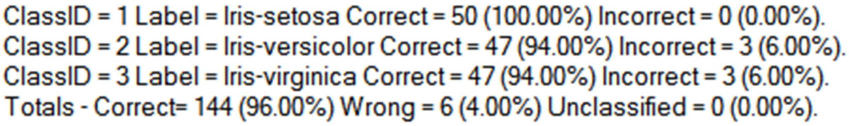
3) petal length in cm; 4) petal width in cm.

with a class label containing three classes:

1) Iris Setosa; 2) Iris Versicolour; 3) Iris Virginica.

The class correlation of the last two is high.

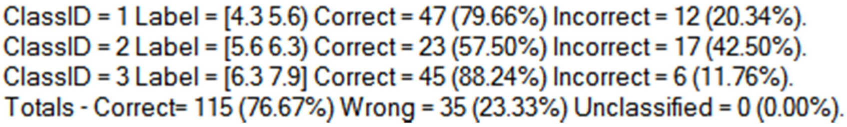
We first use the class attribute to discretize the rest of the attributes and obtain the classification rate by discover*e. The classification rate for the class labels from the data set with labels retained is shown as below:



```
ClassID = 1 Label = Iris-setosa Correct = 50 (100.00%) Incorrect = 0 (0.00%).
ClassID = 2 Label = Iris-versicolor Correct = 47 (94.00%) Incorrect = 3 (6.00%).
ClassID = 3 Label = Iris-virginica Correct = 47 (94.00%) Incorrect = 3 (6.00%).
Totals - Correct= 144 (96.00%) Wrong = 6 (4.00%) Unclassified = 0 (0.00%).
```

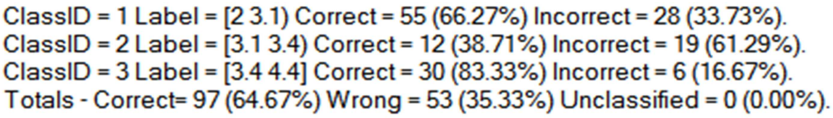
We then remove the class labels from the data set and assume that each of the remaining four as the class attributes (governing attributes) in turn to drive the discretization of all the continuous data and conduct the classification afterward. The classification rate obtained by considering each of the attribute as the governing ones is given below.

Sepal length



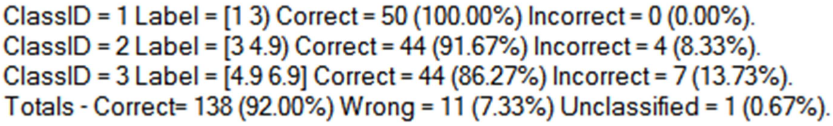
```
ClassID = 1 Label = [4.3 5.6) Correct = 47 (79.66%) Incorrect = 12 (20.34%).
ClassID = 2 Label = [5.6 6.3) Correct = 23 (57.50%) Incorrect = 17 (42.50%).
ClassID = 3 Label = [6.3 7.9] Correct = 45 (88.24%) Incorrect = 6 (11.76%).
Totals - Correct= 115 (76.67%) Wrong = 35 (23.33%) Unclassified = 0 (0.00%).
```

Sepal width



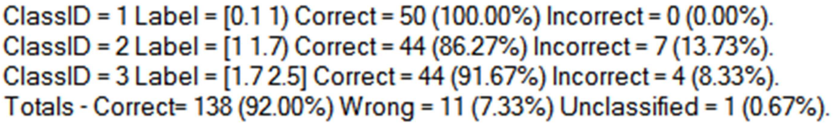
```
ClassID = 1 Label = [2 3.1) Correct = 55 (66.27%) Incorrect = 28 (33.73%).
ClassID = 2 Label = [3.1 3.4) Correct = 12 (38.71%) Incorrect = 19 (61.29%).
ClassID = 3 Label = [3.4 4.4] Correct = 30 (83.33%) Incorrect = 6 (16.67%).
Totals - Correct= 97 (64.67%) Wrong = 53 (35.33%) Unclassified = 0 (0.00%).
```

Petal length*



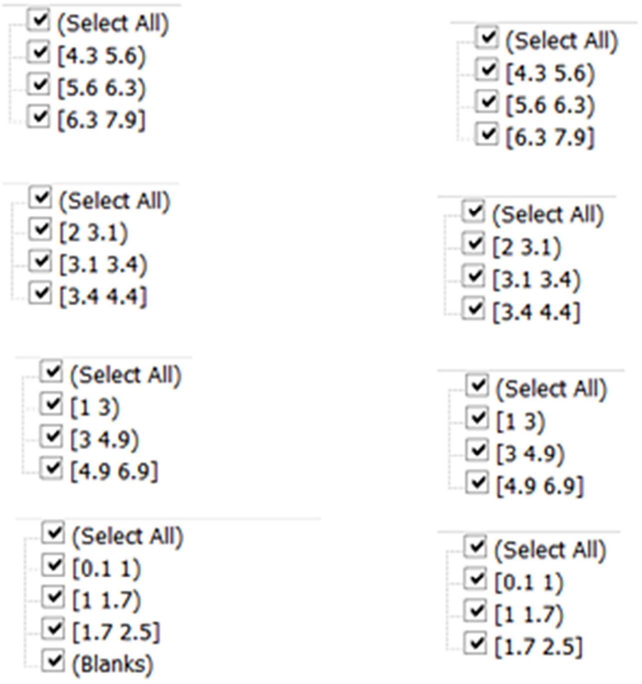
```
ClassID = 1 Label = [1 3) Correct = 50 (100.00%) Incorrect = 0 (0.00%).
ClassID = 2 Label = [3 4.9) Correct = 44 (91.67%) Incorrect = 4 (8.33%).
ClassID = 3 Label = [4.9 6.9] Correct = 44 (86.27%) Incorrect = 7 (13.73%).
Totals - Correct= 138 (92.00%) Wrong = 11 (7.33%) Unclassified = 1 (0.67%).
```

Petal width*



```
ClassID = 1 Label = [0.1 1) Correct = 50 (100.00%) Incorrect = 0 (0.00%).
ClassID = 2 Label = [1 1.7) Correct = 44 (86.27%) Incorrect = 7 (13.73%).
ClassID = 3 Label = [1.7 2.5] Correct = 44 (91.67%) Incorrect = 4 (8.33%).
Totals - Correct= 138 (92.00%) Wrong = 11 (7.33%) Unclassified = 1 (0.67%).
```

From the results obtained, it is clear that the last two attributes could be considered as the governing attributes as they both yield the highest classification rate even without the class labels. To our surprise the discretization results driven by the last attribute is identical to those driven by the class labels as shown below.



☑ (Select All)
☑ [4.3 5.6)
☑ [5.6 6.3)
☑ [6.3 7.9]

☑ (Select All)
☑ [2 3.1)
☑ [3.1 3.4)
☑ [3.4 4.4]

☑ (Select All)
☑ [1 3)
☑ [3 4.9)
☑ [4.9 6.9]

☑ (Select All)
☑ [0.1 1)
☑ [1 1.7)
☑ [1.7 2.5]
☑ (Blanks)

(a)

☑ (Select All)
☑ [4.3 5.6)
☑ [5.6 6.3)
☑ [6.3 7.9]

☑ (Select All)
☑ [2 3.1)
☑ [3.1 3.4)
☑ [3.4 4.4]

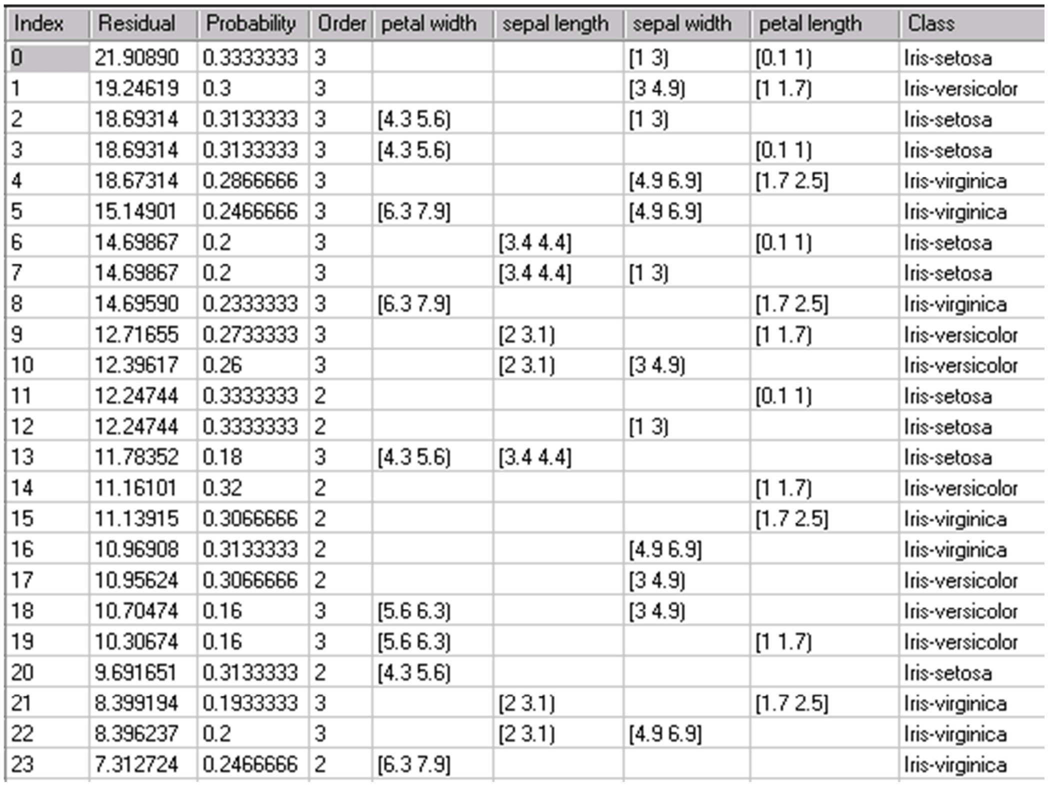
☑ (Select All)
☑ [1 3)
☑ [3 4.9)
☑ [4.9 6.9]

☑ (Select All)
☑ [0.1 1)
☑ [1 1.7)
☑ [1.7 2.5]

(b)

Fig 4.3.1.1 Discretization results of the four attributes: (a) driven by the class label and b) driven by the last attribute when the class labels are taken from the dataset.

Table 4.3.1.1 Examples of Pattern Discovered after the Discretization of the Continuous Data



Index	Residual	Probability	Order	petal width	sepal length	sepal width	petal length	Class
0	21.90890	0.3333333	3			[1 3)	[0.1 1)	Iris-setosa
1	19.24619	0.3	3			[3 4.9)	[1 1.7)	Iris-versicolor
2	18.69314	0.3133333	3	[4.3 5.6)		[1 3)		Iris-setosa
3	18.69314	0.3133333	3	[4.3 5.6)			[0.1 1)	Iris-setosa
4	18.67314	0.2866666	3			[4.9 6.9]	[1.7 2.5]	Iris-virginica
5	15.14901	0.2466666	3	[6.3 7.9]		[4.9 6.9]		Iris-virginica
6	14.69867	0.2	3		[3.4 4.4]		[0.1 1)	Iris-setosa
7	14.69867	0.2	3		[3.4 4.4]	[1 3)		Iris-setosa
8	14.69590	0.2333333	3	[6.3 7.9]			[1.7 2.5]	Iris-virginica
9	12.71655	0.2733333	3		[2 3.1)		[1 1.7)	Iris-versicolor
10	12.39617	0.26	3		[2 3.1)	[3 4.9)		Iris-versicolor
11	12.24744	0.3333333	2				[0.1 1)	Iris-setosa
12	12.24744	0.3333333	2			[1 3)		Iris-setosa
13	11.78352	0.18	3	[4.3 5.6)	[3.4 4.4]			Iris-setosa
14	11.16101	0.32	2				[1 1.7)	Iris-versicolor
15	11.13915	0.3066666	2				[1.7 2.5]	Iris-virginica
16	10.96908	0.3133333	2			[4.9 6.9]		Iris-virginica
17	10.95624	0.3066666	2			[3 4.9)		Iris-versicolor
18	10.70474	0.16	3	[5.6 6.3)		[3 4.9)		Iris-versicolor
19	10.30674	0.16	3	[5.6 6.3)			[1 1.7)	Iris-versicolor
20	9.691651	0.3133333	2	[4.3 5.6)				Iris-setosa
21	8.399194	0.1933333	3		[2 3.1)		[1.7 2.5]	Iris-virginica
22	8.396237	0.2	3		[2 3.1)	[4.9 6.9]		Iris-virginica
23	7.312724	0.2466666	2	[6.3 7.9]				Iris-virginica

Summary: From the experimental results it is obvious that all the questions from (b) to (e) are well answered. In this case the discretization results driven by the governing attributes are identical to those driven by the class labels if they are present.

4.3.2 Mushrooms Data Set (Nominal data)

The mushroom data is a data set consisting of **only nominal data**. It contains 8214 samples with 23 attributes all of the nominal types (Table 4.3.2.1). There are two classes given (edibility e and poisonous p). Since the data set contains of 23 attributes but only two classes, it is used to explore the possibility of the existence of attribute subgroups each of which may govern a certain aspects of the characteristics of the mushrooms. Thus the questions we attempt to answer are related to questions (a), (b), (c) and (e)

Table 4.3.2.1 A Brief Description of the Mushroom Data Set

Attribute Characteristics	No. of Samples	No. of Attributes	No. of Classes
Mixed-Mode Data	8214	23	2

The Mushroom Database is drawn from The Audubon Society Field Guide to North American Mushrooms (1981) by G. H. Lincoff (Pres.), New York: Alfred A. Knopf; Donor: Jeff Schlimmer (Jeffrey.Schlimmer@a.gp.cs.cmu.edu) Date: 27 April 1987. It has been used for concept acquisition by Schlimmer, J.S. (1987). Concept Acquisition Through Representational Adjustment (Technical Report 87-19). Doctoral Dissertation, Department of Information and Computer Science, University of California, Irvine. and extraction of logical rules by Duch W, Adamczak R, Grabczewski K (1996) Extraction of logical rules from training data using back propagation networks, in:
Proc. of the 1st Online Workshop on Soft Computing, 19-30.Aug.1996, pp. 25-30, available on-line at: http://www.bioele.nuee.nagoya-u.ac.jp/wsc1/]

This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family (pp. 500-525). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like ``leaflets three, let it be'' for Poisonous Oak and Ivy. There are 8124 instances and 22 attributes, all nominally valued.

More specifically, the objectives of this experiment are : a) to explore the ranking of the attributes according to their normalized SR2 in the data set with class label included; b) to compare the ranking of the attributes in the data set with class label excluded with the ranking listed in (a); c) to compare the attributes with highest normalized SR2 with the class attributes; d) to show that in a normal setting the attribute with highest normalized SR2 values is also the attribute that render high classification rate if it is considered as a class label instead; e) to show the classificatory characteristics of various attributes; e) to show that significant attribute subgroups exist which can be found by the ACA algorithm; f) to find the mode of each subgroup and compare it with the class attributes to see how representative it is with other attributes in the group. Here we shall report the experimental results

Table 4.3.2.2 shows the ranking of the attributes in the dataset where the class label attribute is included. Here we observe that the ring-type is the mode. Surprisingly, the class attribute is ranked 9th based on the normalized sum of dependence redundancy SR2. This implies that some of the attributes chosen are not necessarily closely related to the class attribute proposed by the biologists.

Table 4.3.2.3 shows the ranking of the attributes according to normalized SR2 from mushroom data after the class label is excluded. Note that the top one remains the same as that in the ranking when class label is included. The second one “stalk root” in Table 4.3.2.3 is ranked fourth in Table 4.3.2.2. The top eight ones in Table 4.3.2.2 remain the same as those in Table 4.3.2.1 indicating the consistence of the governing attributes in relation with the class label attribute.

We next conduct a series of experimental runs treating each of the attribute as the governing one in turn and obtain the classification rate (CR) accordingly. We then rank the attributes according to the classification rates and compare the ranking results with the those ranked according to the normalized SR2 values obtained for the attributes in that group (Table 4.3.2.4).

Table 4.3.2.2 Attributes from mushroom data (with class label included) ranked according to normalized SR2. Note that the class label is not ranked top.

Ranking	**Attributes**	**R1**	**Normalized SR2**
1	ring-type	0.3389	0.136
2	Odor	0.2683	0.1325
3	spore-print-color	0.305	0.124
4	stalk-root	0.2149	0.1198
5	gill-color	0.1547	0.1035
6	stalk-color-above-ring	0.389	0.1034
7	stalk-color-below-ring	0.376	0.1003
8	Population	0.225	0.0857
9	Classes	0.0009	0.0845
10	Habitat	0.1897	0.0839
11	stalk-surface-below-ring	0.3004	0.0838
12	stalk-surface-above-ring	0.3893	0.0816
13	Bruises	0.0207	0.0726
14	cap-color	0.2444	0.0644
15	gill-size	0.1077	0.0613
16	veil-color	0.9019	0.0561
17	gill-attachment	0.8269	0.0552
18	stalk-shape	0.0131	0.0526
19	gill-spacing	0.3621	0.0425
20	ring-number	0.7346	0.0351

21	cap-surface	0.2123	0.0316
22	cap-shape	0.3606	0.03
23	veil-type	1	0

Table 4.3.2.3 Ranking of attributes in mushroom data when the class labels are excluded.

Ranking	**Attributes**	**R1**	**Normalized SR2**
1	**ring-type**	**0.3389**	**0.1357**
2	**stalk-root**	**0.2149**	**0.1231**
3	**spore-print-color**	**0.305**	**0.1215**
4	**Odor**	**0.2683**	**0.1209**
5	**stalk-color-above-ring**	**0.389**	**0.1039**
6	**gill-color**	**0.1547**	**0.1029**
7	**stalk-color-below-ring**	**0.376**	**0.1009**
8	**Population**	**0.225**	**0.0863**
9	**Habitat**	**0.1897**	**0.0855**
10	**stalk-surface-below-ring**	**0.3004**	**0.0817**
11	stalk-surface-above-ring	0.3893	0.0784
12	Bruises	0.0207	0.0709
13	cap-color	0.2444	0.067
14	veil-color	0.9019	0.0578
15	gill-size	0.1077	0.0576
16	gill-attachment	0.8269	0.0572
17	stalk-shape	0.0131	0.0549
18	gill-spacing	0.3621	0.0414
19	ring-number	0.7346	0.0354
20	cap-surface	0.2123	0.0326
21	cap-shape	0.3606	0.0305

22	veil-type	1	0

Table 4.3.2.4 Comparison of Classification Rate (CR) and Normalized SR Ranking of Attributes in Mushroom Data

CR Ranking	SR Ranking	Attributes	Interval #	Distribution	CR (DT)	CR (PD)	Normalized SR2
1	1	ring-type	5	uneven	100	98.15	0.1357
2	**2**	**stalk-root**	**5**	**Even**	**100**	**85.28**	**0.1231**
3	12	Bruises	2	Even	100	100	0.0709
4	15	gill-size	2	Skew	100	98.38	0.0576
5	17	stalk-shape	2	Even	100	98.38	0.0549
6	19	ring-number	3	Biased	100	92.17	0.0354
7	16	gill-attachment	2	Biased	99.78	97.54	0.0572
8	14	veil-color	4	Biased	98.92	97.54	0.0578
9	18	gill-spacing	2	Skew	98.82	97.42	0.0414
10	4	Odor	9	uneven	80.9	67.26	0.1209
11	10	stalk-surface-below-ring	4	normal	80.8	74.35	0.0817
12	11	stalk-surface-above-ring	4	Even	80.8	79.22	0.0784
13	3	spore-print-color	5	uneven	74.59	61.88	0.1215
14	9	Habitat	6	uneven	66.96	51.65	0.0855
15	8	Population	6	uneven	63.76	55.15	0.0863
16	5	stalk-color-above-ring	9	uneven	63.37	58.2	0.1039
17	7	stalk-color-below-ring	9	uneven	63.17	57.21	0.1009
18	20	cap-surface	4	uneven	55.29	52.72	0.0326
19	21	cap-shape	6	uneven	45.49	31.02	0.0305
20	6	gill-color	12	uneven	45.42	26.98	0.1029

21	13	cap-color	10	uneven	44.26	39.03	0.067
22	22	veil-type	NA	NA	NA	NA	0

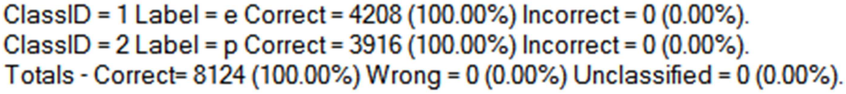
First we observe in Table 4.3.2.2 that in the SR2 ranking, the two attributes, the ring-type and stalk-root top all other attributes. They are ranked first and fourth in Table 4.3.2.1 when the class labels are present. That the ranking of the Class Attribute is not ranked top according to SR2 indicates that its interdependence w with all the other attributes in the group may not be the highest. Rather, the two other attributes, the ring-type and stalk-root are more governing in the sense that they have higher interdependence with other attributes in the group.

We then conduct classification experiments on these two sets of data. We first conduct supervised classification of the data according to the class labels given and obtain 100% rate of correct classification (Figure 4.3.2.1(a)). We then move on to classify the same set of data with the class label removed. In the first classification run, we assume that the ring-type would serve as the governing attribute, i.e. it is treated as the class label in the supervised classification run, and again a 100% of the classification rate is obtained (Figure 4.3.2.1(b)). We next take “stalk root” as the governing attribute and again obtain 100% classification rate (Figure 4.3.2.1 (c)). Though the two sets of the classification details may not be exactly the same, their strong correlation with rest of attributes indicates they both have some governing characteristics as reflected by their high classification (i.e. feature-class dependence) rate.

To address the issues that the class label is not ranked top according to its normalized SR2, we make the following observations. As pointed in the reference source, the Guide clearly states that there is no simple rule for determining the edibility of a mushroom. Furthermore, the biologists also place the last two classes of unknown edibility and not recommended into the poisonous category. This means that there could be more subtle attributes that govern the intrinsic classes. To explore the characteristic of the proposed classification scheme, we will conduct the ACA on the set of 23 attributes and see whether or not they might be better grouped into subgroups each of which might characterize certain aspects of the mushroom characteristics.

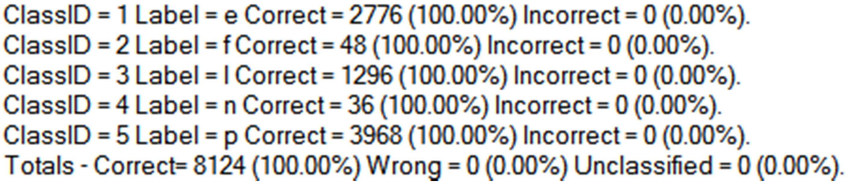
In our attribute clustering experiments, we will apply ACA first to the data set with class label and then with that without class labels. We will compare the results so as to gain insight into the class labels and the intrinsic governing attribute issues.

Table 4.3.2.4 gives the attribute groups discovered in the first experiments. This is the result of the local optimal solution. In the first cluster we observe that the class labels are more closely related to the odor, gill-size, cap-color and the ring-number of mushrooms. Note that apart from odor which is ranked 4th, the SR2 ranking of the rest of the three attributes in the group are not too high (cap-color ranked 13th, gill size 15th and ring-number 19th). It shows that as far as the “edibility” and “poisonous” properties are concern, these four attributes are most relevant. The others may have various interdependence characteristics to pull them together into more correlated groups. This is an important aspect we should seriously consider if there is no obvious class labels are available. Unless we have full knowledge ahead of time, for a given of data we should explore its internal association before a meaningful analysis could be sorted out. This is also an important objective for the proposed methodology, especially designed for situations when class information is lacking.



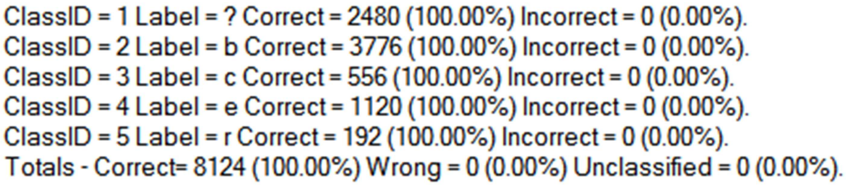
```
ClassID = 1 Label = e Correct = 4208 (100.00%) Incorrect = 0 (0.00%).
ClassID = 2 Label = p Correct = 3916 (100.00%) Incorrect = 0 (0.00%).
Totals - Correct= 8124 (100.00%) Wrong = 0 (0.00%) Unclassified = 0 (0.00%).
```

(a) Classification rate of Class Labels



```
ClassID = 1 Label = e Correct = 2776 (100.00%) Incorrect = 0 (0.00%).
ClassID = 2 Label = f Correct = 48 (100.00%) Incorrect = 0 (0.00%).
ClassID = 3 Label = l Correct = 1296 (100.00%) Incorrect = 0 (0.00%).
ClassID = 4 Label = n Correct = 36 (100.00%) Incorrect = 0 (0.00%).
ClassID = 5 Label = p Correct = 3968 (100.00%) Incorrect = 0 (0.00%).
Totals - Correct= 8124 (100.00%) Wrong = 0 (0.00%) Unclassified = 0 (0.00%).
```

(b) Classification rate of Ring-Type



```
ClassID = 1 Label = ? Correct = 2480 (100.00%) Incorrect = 0 (0.00%).
ClassID = 2 Label = b Correct = 3776 (100.00%) Incorrect = 0 (0.00%).
ClassID = 3 Label = c Correct = 556 (100.00%) Incorrect = 0 (0.00%).
ClassID = 4 Label = e Correct = 1120 (100.00%) Incorrect = 0 (0.00%).
ClassID = 5 Label = r Correct = 192 (100.00%) Incorrect = 0 (0.00%).
Totals - Correct= 8124 (100.00%) Wrong = 0 (0.00%) Unclassified = 0 (0.00%).
```

(c) Classification rate of Stalk-Root

Figure 4.3.2.1 Classification rate of the induced intervals of the governing attributes using method of decision tree C.40.

(a) Classification rate of mushroom data based on the given class label.

(b) Classification rate of an assumed “governing attribute” ring-type from mushroom data after the class label is excluded;

(c) Classification rate of another assumed “governing attribute” stalk-root from the mushroom data after the class label is excluded.

From the comparison results tabulated in Table 4.3.2.4, and Figure 4.3.2.1 (b) and (c), it seems that as far as the distribution of the categorical values is concerned, stalk-root has a more even distribution in charactering the data without class label.

Table 4.3.2.5 shows the results of attribute clustering of the data set without class labels by ACA. Note that the optimal attribute cluster configuration consists of two clusters, one headed by the mode ring-type and the other by the mode stalk-root. When we look into the characteristics of these two governing attributes, we observe in Table 4.3.2.4 that although the SR2 value for ring-type is a little higher, yet the distribution of the categories it encompassed is less even when comparing the classification rate of their categories from Figure 4.3.2.1 (b) and (c) . Thus as far as the representative characteristic of these two attribute in the attribute groups is concerned, the latter seems to offer a better candidate. This will be explored by our future research.

A close look at the attributes forming these two correlated groups, we note that all the attributes associated with the class label (Table 4.3.2.5) reside in the second group headed by the mode of stalk-root. That means that this group should provide better correlated attributes with the classes of edibility and poisonous. This kind of insights for the analysis and the understanding of a large database with no or little class information could be effectively provided by 1) our ACA, 2) our mode finding algorithm and 3) our governing attribute driven discretization and classification procedure presented in this dissertation.

Table 4.3.2.5 Attribute Clusters of Mushroom Data with class label included. Three cluster configurations are the optimal. They are tabulated with the attributes in each cluster ranked according to the normalized value of the attribute of the group.

Attributes	**R1**	**Normalized SR2**
Odor	0.2683	0.1823
Classes	0.0009	0.1381
gill-size	0.1077	0.0993
cap-color	0.2444	0.0571
ring-number	0.7346	0.0356

Attributes	**R1**	**Normalized SR2**
ring-type	0.3389	0.2157
spore-print-color	0.305	0.1596
stalk-color-above-ring	0.389	0.1417
stalk-surface-above-ring	0.3893	0.1407
stalk-surface-below-ring	0.3004	0.1406
stalk-color-below-ring	0.376	0.1382
gill-color	0.1547	0.1284
Bruises	0.0207	0.1184
stalk-shape	0.0131	0.0758

Attributes	**R1**	**Normalized SR2**
stalk-root	0.2149	0.1359
population	0.225	0.1265

Habitat	0.1897	0.1086
gill-spacing	0.3621	0.0667
cap-surface	0.2123	0.05
cap-shape	0.3606	0.0422

Table 4.3.2.6 Attribute Clusters of Mushroom Data with class label excluded. Two cluster configuration is the optimal. They are tabulated with the attributes in each cluster ranked according to the normalized value of the attribute of the group.

Attributes	**R1**	**Normalized SR2**
ring-type	0.3389	0.2157
spore-print-color	0.305	0.1596
stalk-color-above-ring	0.389	0.1417
stalk-surface-above-ring	0.3893	0.1407
stalk-surface-below-ring	0.3004	0.1406
stalk-color-below-ring	0.376	0.1382
gill-color	0.1547	0.1284
Bruises	0.0207	0.1184
stalk-shape	0.0131	0.0758

Attributes	R1	Normalized SR2
stalk-root	0.2149	0.1352
Odor	0.2683	0.1113
population	0.225	0.1087
Habitat	0.1897	0.1007
cap-color	0.2444	0.0695
gill-size	0.1077	0.067
gill-spacing	0.3621	0.0527
cap-surface	0.2123	0.0395

cap-shape	0.3606	0.0382
ring-number	0.7346	0.0377

.

Summary: The experimental results show that in order to have an in-depth understanding of a large dataset, it is beneficial to go through the attribute clustering process. The attribute clustering and the identification of modes (or other top governing attributes) in the original data set and also the clustered attribute groups render considerable insights into the inherent makeup of the data and the problems they reflect**. In the situation when no class label is available, the mode in the dataset and in each of the attribute cluster can be considered as the most representative or the governing one.**

4.3.3 Adult Data Set (Mixed Mode Data)

This database was extracted from the census bureau database found at (Table 4.3.3.1) by http://www.census.gov/ ftp/pub/DES/www/welcome.html. It contains 48842 instances of mix of continuous and discrete data with 14 attributes (Table 4.3.3.2). It has been used for predictive whether a person makes over 50k a year or not. We use this mixed-mode data set to answer the questions (a) to (e). More specifically, the experiment is used: 1) to demonstrate the existence of attribute subgroups in the mixed-mode data set; 2) to illustrate the attainment of attribute cluster configuration and the grouping of cluster items in situations with or without class label; 3) to show the classification characteristics of various attributes in different attribute groups found by ACA; 4) to show that the attribute with highest normalized SR, or simply the mode, in the attribute group is usually with high classification rate if it is assumed to take the role of a class label. The experiment results show that the mode in each attribute group/cluster can be considered as the most discriminative/representative or governing attribute to drive the discretization of continuous attributes in the attribute group/cluster.

In this experiment, the proposed method is used to calculate the normalized mutual information, *R*, among the attributes. Their values are tabulated in Table 4.3.3.3 for the data set with class label excluded and in Table 4.3.3. 4 with class label included.

Based on the *R* values, our ACA found the optimal cluster configuration in the given data set. Table 4.3.3.3, Table 4.3.3.4 and Table 4.3.3.5 reports the value of the sum of significant MR calculated during the clustering process. It is obvious that 3 attribute clusters and 5 attribute clusters are local optimal for the data with the class label excluded and those with the class label included respectively. In this regards, we compare the attribute items and the modes in each of the attribute clusters in Table 4.3.3.6

Table 4.3.3.1 A Brief Description of Adult Data Set

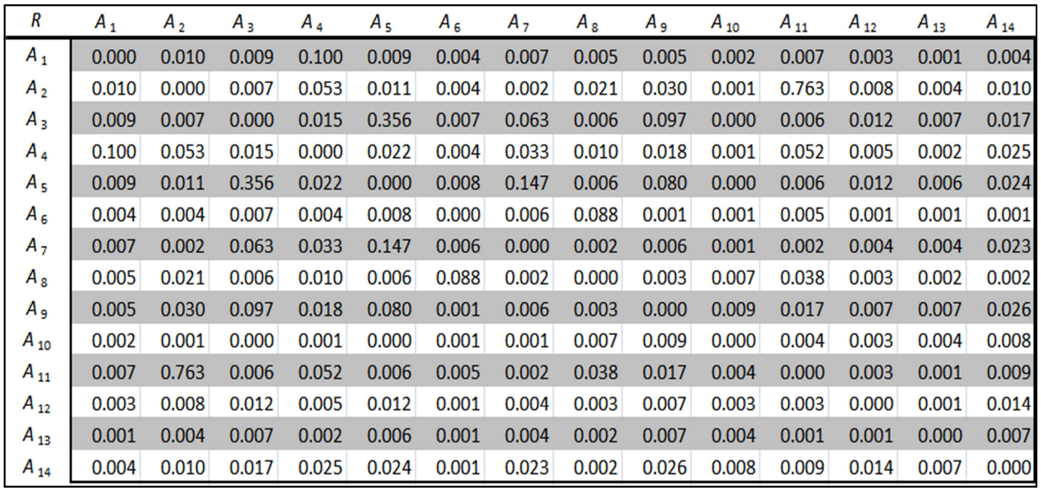
Data Description				
Data Set	Attribute Characteristics	No. of Samples	No. of Attributes	No. of Classes
Adult	Mixed-Mode Data	48842	14	2

Table 4.3.3.2 The Attributes of Adult Data Set

Attribute	Name	Characteristics
A_1	Work class	Discrete
A_2	Education	
A_3	marital-status	
A_4	Occupation	
A_5	Relationship	
A_6	Race	
A_7	Sex	
A_8	native-country	
A_9	Age	Continuous
A_{10}	Fnlwgt	
A_{11}	education-num	

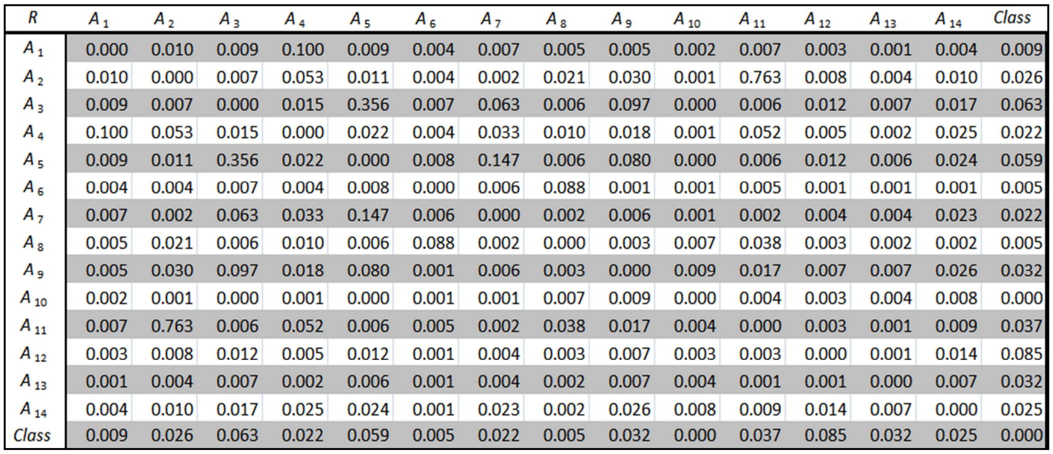
A_{12}	capital-gain	
A_{13}	capital-loss	
A_{14}	hours-per-week	
Class	Income	Discrete

Table 4.3.3.3 Normalized Mutual Information between Attributes of Adult Data Set with Class



R	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}
A_1	0.000	0.010	0.009	0.100	0.009	0.004	0.007	0.005	0.005	0.002	0.007	0.003	0.001	0.004
A_2	0.010	0.000	0.007	0.053	0.011	0.004	0.002	0.021	0.030	0.001	0.763	0.008	0.004	0.010
A_3	0.009	0.007	0.000	0.015	0.356	0.007	0.063	0.006	0.097	0.000	0.006	0.012	0.007	0.017
A_4	0.100	0.053	0.015	0.000	0.022	0.004	0.033	0.010	0.018	0.001	0.052	0.005	0.002	0.025
A_5	0.009	0.011	0.356	0.022	0.000	0.008	0.147	0.006	0.080	0.000	0.006	0.012	0.006	0.024
A_6	0.004	0.004	0.007	0.004	0.008	0.000	0.006	0.088	0.001	0.001	0.005	0.001	0.001	0.001
A_7	0.007	0.002	0.063	0.033	0.147	0.006	0.000	0.002	0.006	0.001	0.002	0.004	0.004	0.023
A_8	0.005	0.021	0.006	0.010	0.006	0.088	0.002	0.000	0.003	0.007	0.038	0.003	0.002	0.002
A_9	0.005	0.030	0.097	0.018	0.080	0.001	0.006	0.003	0.000	0.009	0.017	0.007	0.007	0.026
A_{10}	0.002	0.001	0.000	0.001	0.000	0.001	0.001	0.007	0.009	0.000	0.004	0.003	0.004	0.008
A_{11}	0.007	0.763	0.006	0.052	0.006	0.005	0.002	0.038	0.017	0.004	0.000	0.003	0.001	0.009
A_{12}	0.003	0.008	0.012	0.005	0.012	0.001	0.004	0.003	0.007	0.003	0.003	0.000	0.001	0.014
A_{13}	0.001	0.004	0.007	0.002	0.006	0.001	0.004	0.002	0.007	0.004	0.001	0.001	0.000	0.007
A_{14}	0.004	0.010	0.017	0.025	0.024	0.001	0.023	0.002	0.026	0.008	0.009	0.014	0.007	0.000

Table 4.3.3.5 Normalized Mutual Information between Attributes of Adult Data Set with Class Label included



R	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	*Class*
A_1	0.000	0.010	0.009	0.100	0.009	0.004	0.007	0.005	0.005	0.002	0.007	0.003	0.001	0.004	0.009
A_2	0.010	0.000	0.007	0.053	0.011	0.004	0.002	0.021	0.030	0.001	0.763	0.008	0.004	0.010	0.026
A_3	0.009	0.007	0.000	0.015	0.356	0.007	0.063	0.006	0.097	0.000	0.006	0.012	0.007	0.017	0.063
A_4	0.100	0.053	0.015	0.000	0.022	0.004	0.033	0.010	0.018	0.001	0.052	0.005	0.002	0.025	0.022
A_5	0.009	0.011	0.356	0.022	0.000	0.008	0.147	0.006	0.080	0.000	0.006	0.012	0.006	0.024	0.059
A_6	0.004	0.004	0.007	0.004	0.008	0.000	0.006	0.088	0.001	0.001	0.005	0.001	0.001	0.001	0.005
A_7	0.007	0.002	0.063	0.033	0.147	0.006	0.000	0.002	0.006	0.001	0.002	0.004	0.004	0.023	0.022
A_8	0.005	0.021	0.006	0.010	0.006	0.088	0.002	0.000	0.003	0.007	0.038	0.003	0.002	0.002	0.005
A_9	0.005	0.030	0.097	0.018	0.080	0.001	0.006	0.003	0.000	0.009	0.017	0.007	0.007	0.026	0.032
A_{10}	0.002	0.001	0.000	0.001	0.000	0.001	0.001	0.007	0.009	0.000	0.004	0.003	0.004	0.008	0.000
A_{11}	0.007	0.763	0.006	0.052	0.006	0.005	0.002	0.038	0.017	0.004	0.000	0.003	0.001	0.009	0.037
A_{12}	0.003	0.008	0.012	0.005	0.012	0.001	0.004	0.003	0.007	0.003	0.003	0.000	0.001	0.014	0.085
A_{13}	0.001	0.004	0.007	0.002	0.006	0.001	0.004	0.002	0.007	0.004	0.001	0.001	0.000	0.007	0.032
A_{14}	0.004	0.010	0.017	0.025	0.024	0.001	0.023	0.002	0.026	0.008	0.009	0.014	0.007	0.000	0.025
Class	0.009	0.026	0.063	0.022	0.059	0.005	0.022	0.005	0.032	0.000	0.037	0.085	0.032	0.025	0.000

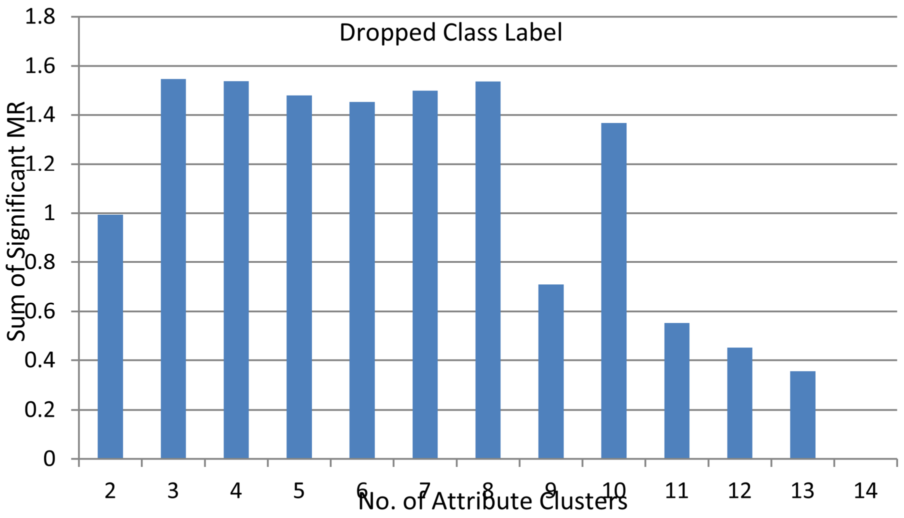
In our proposed method, no class information is required; nevertheless, the results reported in Table 4.3.3.9 shows that even without class information, our proposed method and ACA are able to group interdependent attributes together. This demonstrates the effectiveness of our method to extract the same intrinsic information inherent in the classes.

Table 4.3.3.6 The Sum of Significant MR obtained for each *k* of the *k*-Mode ACA

No. of Attribute Cluster, *k*	Excluded Class Label	Included Class Label
	Sum of Significant MR	Sum of Significant MR
2	0.993065	1.559977
3	*1.546628	1.599815
4	1.536047	1.597278
5	1.478268	*1.685009
6	1.452389	1.603821
7	1.498127	1.504654
8	1.53544	1.522862
9	0.708419	1.032613
10	1.366747	1.393317
11	0.553327	0.914691
12	0.452937	0.553327
13	0.355844	0.452937
14	0	0.763257
15	-	0

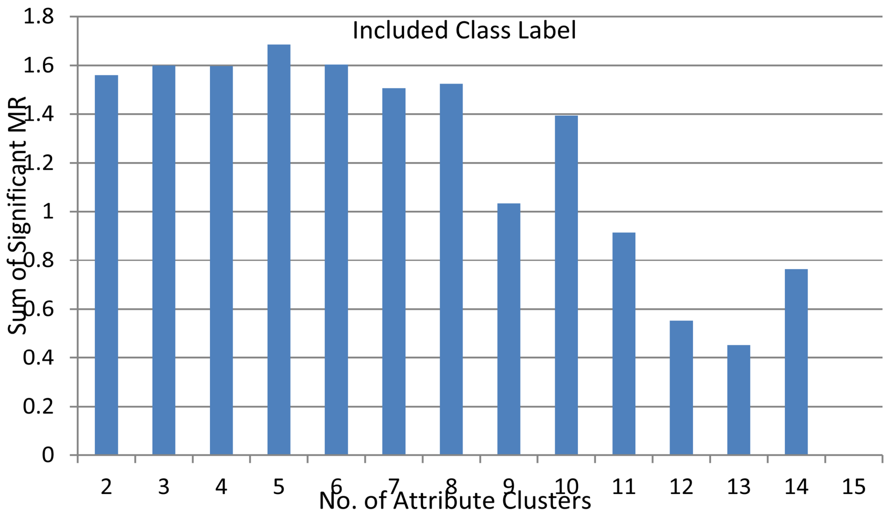
* Highest Sum of Significant MR Implies Optimal *k* =3 for Data Set Dropped Class Label and Optimal *k* = 5 for Data Set Included Class Label.

Table 4.3.3.7. The Plot of the Sum of Significant MR (with class label dropped)



* Highest Sum of Significant MR Implies Optimal *k* =3.

Table 4.3.3.8 The Plot of the Sum of Significant MR (with class label included)



* Highest Sum of Significant MR Implies Optimal *k* =5.

Table 4.3.3.9. The Attribute Clusters and their Mode Obtained by ACA

Attribute Cluster Items		
Attribute Group	**Dropped Class Label**	**Included Class Label**
1	*native-country, race, fnlwgt	*native-country, race, fnlwgt
2	*education, workclass, occupation, education-num	*education-num, education

3	*relationship, marital-status, sex, age, capital-gain, capital-loss, hours-per-week	*relationship, marital-status, sex, age
4	-	*workclass, occupation
5	-	*income (class), capital-gain, capital-loss, hours-per-week

* The attribute marked with "*" is the mode of the attribute group. A mode is with the highest normalized mutual information in the attribute group.

To further investigate the attributes resided in each attribute group, we study the classificatory aspect of them to show that in a normal setting the mode is also the attribute that renders good enough classification rate if it is regarded as a class label. The attribute clusters, normalized SR values and their classification performance are tabulated in Table 4.3.3.9.

Table 4.3.3.9 Attribute Clusters of Adult Data with Class Label Excluded with their Normalized SR Values and their Classification Accuracy by PD with a 95% Confidence Interval.

Attribute	Characteristics	Normalized SR	Classification Accuracy (%)
* native-country	Discrete	0.0952	89.59
race	Continuous	0.0898	84.43
fnlwgt	Continuous	0.0083	5.41

* The attribute marked with "*" is the mode of the attribute group. A mode is with the highest normalized mutual information in the attribute group.

Attribute	Characteristics	Normalized SR	Classification Accuracy (%)
* education	Discrete	0.8263	71.09
workclass	Discrete	0.8218	57.69
occupation	Discrete	0.2051	20.94
education-num	Continuous	0.1173	-

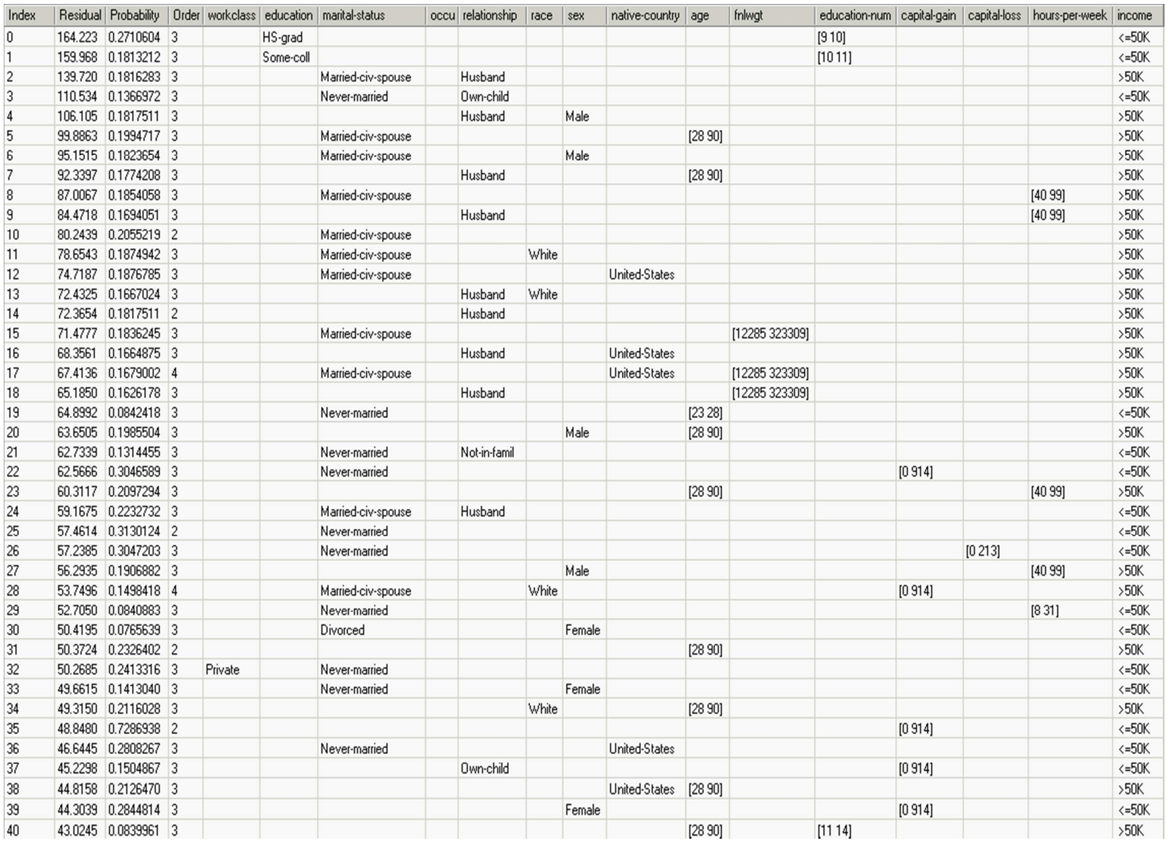
* The attribute marked with "*" is the mode of the attribute group. A mode is with the highest

normalized mutual information in the attribute group.

Attribute	Characteristics	Normalized SR	Classification Accuracy (%)
* relationship	Discrete	0.6251	72
# marital-status	Discrete	0.5525	74.78
sex	Discrete	0.2465	68.95
age	Continuous	0.2229	-
^ capital-gain	Continuous	0.1100	99.51
^ capital-loss	Continuous	0.0495	95.33
hours-per-week	Continuous	0.0313	14.54

* The attribute marked with "*" is the mode of the attribute group. A mode is with the highest normalized mutual information in the attribute group. ^ The attribute marked with "^" implies the data is sparse. # The attribute marked with "#" holds the highest classification accuracy, even higher than the mode.

Table 4.3.3.10 Pattern Discovered



Index	Residual	Probability	Order	workclass	education	marital-status	occu	relationship	race	sex	native-country	age	fnlwgt	education-num	capital-gain	capital-loss	hours-per-week	income
0	164.223	0.2710604	3		HS-grad									[9 10]				<=50K
1	159.968	0.1813212	3		Some-coll									[10 11]				<=50K
2	139.720	0.1816283	3			Married-civ-spouse		Husband										>50K
3	110.534	0.1366972	3			Never-married		Own-child										<=50K
4	106.105	0.1817511	3					Husband		Male								>50K
5	99.8863	0.1994717	3			Married-civ-spouse						[28 90]						>50K
6	95.1515	0.1823654	3			Married-civ-spouse				Male								>50K
7	92.3397	0.1774208	3					Husband				[28 90]						>50K
8	87.0067	0.1854058	3			Married-civ-spouse											[40 99]	>50K
9	84.4718	0.1694051	3					Husband									[40 99]	>50K
10	80.2439	0.2055219	2			Married-civ-spouse												>50K
11	78.6543	0.1874942	3			Married-civ-spouse			White									>50K
12	74.7187	0.1876785	3			Married-civ-spouse					United-States							>50K
13	72.4325	0.1667024	3					Husband	White									>50K
14	72.3654	0.1817511	2					Husband										>50K
15	71.4777	0.1836245	3			Married-civ-spouse							[12285 323309]					>50K
16	68.3561	0.1664875	3					Husband			United-States							>50K
17	67.4136	0.1679002	4			Married-civ-spouse					United-States		[12285 323309]					>50K
18	65.1850	0.1626178	3					Husband					[12285 323309]					>50K
19	64.8992	0.0842418	3			Never-married						[23 28]						<=50K
20	63.6505	0.1985504	3							Male		[28 90]						>50K
21	62.7339	0.1314455	3			Never-married		Not-in-famil										<=50K
22	62.5666	0.3046589	3			Never-married									[0 914]			<=50K
23	60.3117	0.2097294	3									[28 90]					[40 99]	>50K
24	59.1675	0.2232732	3			Married-civ-spouse		Husband										<=50K
25	57.4614	0.3130124	2			Never-married												<=50K
26	57.2385	0.3047203	3			Never-married										[0 213]		<=50K
27	56.2935	0.1906882	3							Male							[40 99]	>50K
28	53.7496	0.1498418	4			Married-civ-spouse			White						[0 914]			>50K
29	52.7050	0.0840883	3			Never-married											[8 31]	<=50K
30	50.4195	0.0765639	3			Divorced				Female								<=50K
31	50.3724	0.2326402	2									[28 90]						>50K
32	50.2685	0.2413316	3	Private		Never-married												<=50K
33	49.6615	0.1413040	3			Never-married				Female								<=50K
34	49.3150	0.2116028	3						White			[28 90]						>50K
35	48.8480	0.7286938	2												[0 914]			<=50K
36	46.6445	0.2808267	3			Never-married					United-States							<=50K
37	45.2298	0.1504867	3					Own-child							[0 914]			<=50K
38	44.8158	0.2126470	3								United-States	[28 90]						>50K
39	44.3039	0.2844814	3							Female					[0 914]			<=50K
40	43.0245	0.0839961	3									[28 90]		[11 14]				>50K

4.3.4 Colon Cancer Data Set (Continuous Data)

The *colon-cancer* dataset consists of 62 samples and 2,000 genes, which is represented by a 2,000 × 62 expression table. The samples are composed of tumor biopsies collected from tumors and normal biopsies collected from healthy part of the colons of the same patient. Each sample has been pre-classified into one of the two classes: *normal* and *cancer*. This set of data is less explicit and difficult to explain. It is large in the sense that it contains 2,000 genes which can be treated as attributes with the gene expression of continuous values as their outcomes (Table 4.3.4.1). In [21], the researchers treated the data as categorical data by first discretizing the continuous values into intervals based on the class labels (cancerous and normal patients) given. The fundamental problems of mode finding and attribute clustering notions have not been solved. In this dissertation, we have developed an algorithm which solves both the mode finding and the attribute clustering for mixed-mode data. In this experiment, we apply our algorithm on the original continuous data to see whether or not we could achieve the same goal even without the knowledge of the class labels. It then gives us a solid base of comparison and further affirms the validity of our approach. Using the same set of original data with class labels excluded, we try to answer questions (a) – (e). We are particularly interested to find out how effective are the discrete intervals obtained for all the gene expressions based on the governing genes in classifying the cancer and normal genes.

Table 4.3.4.1. Colon Cancer Data Set

Data Description				
Data Set	Attribute Characteristics	No. of Samples	No. of Attributes	No. of Classes
Colon Cancer	Continuous Data	62	2000	2

For demonstrative purpose, we select top 5 attributes of 2 clusters of colon cancer data

(Table 4.3.4.2) as found and reported in [21]. First we used the numerical method to compute the normalized mutual information (R) between continuous attributes. The R results for all the gene pairs are tabulated in Table 4.3.4.3. Based on the R values, our ACA algorithm is able to find that the two attribute clusters as the optimal cluster configuration corresponding to the result found in [21] which assumes that class labels are given in their attribute clustering. Table 4.3.4.4. displays the value of the sum of significant MR calculated during the clustering process. Two clusters configuration is obviously the optimal local solution. In the work reported in this thesis, no class label information is taken into account. The two attribute clusters obtained are given in Table 4.3.4.5. This demonstrates the effectiveness of our method in getting the same *intrinsic class information* and *gene grouping information* inherent in the data.

In response to question (e), we explore how well the performance of our discretization results is when the partitioned intervals are used as associative events in the classification. We now take all the seven attribute clusters found in [21] and discretize all the continuous attributes by the governing attributes (modes) found in each of the clusters. We then pooled 5 discretized attributes from each of the 7 found attribute clusters together to obtain a data set of 35 attributes. We refer this set as a "selected attribute pool of most representative attributes". We then apply classification of the gene tissue class using the discretized intervals obtained without relying on class labels.

Table 4.3.4.2. The Selected Top Five Attributes of the Two Clusters Found in the Colon Cancer Data Set reported in [21].

Attribute	Ra	Attrib	Accession
Clusters	nk	ute	Number
1	1	A_1	H05814
	2	A_2	X02874
	3	A_3	U33429
	4	A_4	H22579

	5	A_5	H25940
2	1	A_6	T73092
	2	A_7	R26146
	3	A_8	T90851
	4	A_9	R93337
	5	A_{10}	T69446

Table 4.3.4.3. Normalized Mutual Information between Attributes of the Selected 10 Continuous Attributes

R	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
A_1	0	0.2202	0.2011	0.3521	0.2905	0.0946	0.1072	0.1147	0.0809	0.1038
A_2	0.2202	0	0.1425	0.2073	0.1821	0.123	0.0902	0.1058	0.076	0.1466
A_3	0.2011	0.1425	0	0.17	0.1715	0.0733	0.0467	0.0339	0.0738	0.0669
A_4	0.3521	0.2073	0.17	0	0.2426	0.0856	0.1398	0.1053	0.0816	0.1138
A_5	0.2905	0.1821	0.1715	0.2426	0	0.119	0.0752	0.0848	0.1045	0.1065
A_6	0.0946	0.123	0.0733	0.0856	0.119	0	0.2248	0.1445	0.1635	0.4401
A_7	0.1072	0.0902	0.0467	0.1398	0.0752	0.2248	0	0.1391	0.2095	0.2269
A_8	0.1147	0.1058	0.0339	0.1053	0.0848	0.1445	0.1391	0	0.177	0.1285
A_9	0.0809	0.076	0.0738	0.0816	0.1045	0.1635	0.2095	0.177	0	0.1439
A_{10}	0.1038	0.1466	0.0669	0.1138	0.1065	0.4401	0.2269	0.1285	0.1439	0

Table 4.3.4.4 The Plot of the Sum of Significant MR obtained for each *k* of the *k*-Mode ACA Algorithm.

No. of Attribute Clusters, *k*	Sum of Significant MR
*2	2.0368

3	1.9079
4	1.7309
5	1.3972
6	1.0883
7	1.0124
8	0.7922
9	0.3521
10	0

* Highest Sum of Significant MR Implies Optimal *k* =2.

Table 4.3.4.6 The Attribute Clusters and their Mode Obtained by ACA.

Cluster	Mode	Significant MR	Item
1	A_1	1.0639	A_1, A_4, A_5, A_2, A_3
2	A_6	0.9729	A_6, A_{10}, A_7, A_9, A_8

* The found cluster items and modes are the same as [21]

Since the class label for the Colon Cancer data set is known, we can make use of this ground truth to devise an evaluation scheme in accessing the performance of different discretization methods. First, we apply different discretization techniques on the selected attribute pool. We then run classification experiments on the discretized selected attribute pool to obtain classification results. To compare our proposed discretization with others, the benchmark results reported by [21] are given. In the classification performance evaluation process, the Leave-One-Out Cross Validation (LOOCV) which is the same validation method employed by [21] is adopted. Applying LOOCV in the Colon Cancer data set, the 1st sample is selected as the testing set and the remaining 61 samples are selected as the training test. This procedure repeats from 1st sample to 62nd sample. The classification accuracy is computed as the overall number of correct classification from the 62 iterations, divided by the total number of samples in the data, which in this set is 62.

From the experiment results shown in table 4.3.4.7, it is interesting to remark that the classification result of the colon cancer data discretized by the proposed discretization is close to those discretized by OCDD which makes use of class information.

In the classification experiment, it demonstrates that our proposed discretization approach enables an inductive learning algorithm to build an accurate classifier which achieves competitive classification result to one using class label information in the discretization of the continuous attributes in the database.

Table 4.3.4.7. The Classification Performance of C5.0 on the Attribute Pools Selected by Different Attribute Clustering Techniques and Discretized by Different Discretization Techniques in the Colon Cancer Data Set

Classification Performance	**The Proposed Discretization**	**OCDD**					
	ACA	**ACA**	***t*-value**	***k*-means**	**SOM**	**Biclustering**	**MRMR**
Classification Accuracy (%)	**88.71**	**91.9**	**74.2**	**71.0**	**43.5**	**75.8**	**83.9**

Summary: In the colon cancer experiment, we show that our ACA could cluster attributes as effective as one which has taken class labels into account. It also shows that both the attribute clustering and repooling process work for the set of discretized data effectively by our proposed method. More surprisingly is that the use of the discretized results obtained based on our discovered governing gene will produce discretized intervals that enable our classificatory system to achieve high classification rate of cancerous and normal patients equivalent to systems using class labels. The results of this experiment implies that our governing attribute driven discretization scheme does effectively use some of the class information inherent in the data to achieve high classification rate.

4.4. Experiments on Meteorological (MET) Database

The meteorological (MET) database is a large database consisting of 44 attributes and 8784 samples. The MET data was taken from 5 different surface stations over a one-year-long period (8760 recorders) in the great urban region of Guangzhou City, Guangdong province, China, within about WE-200km and NS-300km (Figure 4.4.1). The types of the meteorological parameters (attributes) collected from the surface stations include 6 discrete attributes and 25 continuous attributes. The five surface stations denoted by the alphabets S =A, B, C, D, E are stations as listed below.

Station A =Guangzhou Metropolis;
Station B =Foshan City;   Station C =Shenzhen City;
Station D = Dongguan City;   Station E =Zhongshan City;

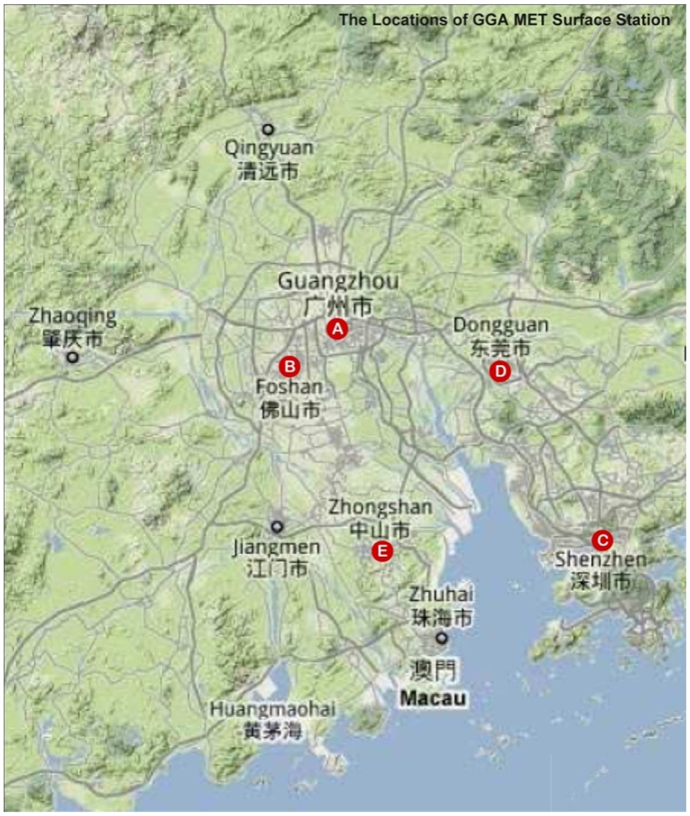


Figure 4.4.1 Guangzhou Urban Region (GGA)

Table 4.4.1 gives a brief description of the MET Data. In the MET database, there are totally 43 attributes where 18 of them are categorical attributes and 25 are continuous attributes. They are listed in Table M- Ii as shown below with the value types given inside the respective brackets.

Table 4.4.1. MET Data Description

Data Description				
Data Set	Attribute Characteristics	No. of Samples	No. of Attributes	No. of Classes
MET DATA (GGA)	**Mixed-Mode Data**	8784	44 D19 & C25	Unknown

Table 4.4.2 Attributes and attribute values in the MET database.

Attr.	name	D/C mode	Notes
MM	Month	Discrete	Month
DD	Day	Discrete	Day
HH	Hour	Discrete	Hour
S1	TC	Discrete	Total Cloudiness
S2	LC	Discrete	Lower Cloudiness
S3	DBT	Continuous	Dry Bulb Temperature
S4	DPT	Continuous	Dew Point Temperature
S5	RH	Continuous	Relative Humidity
S6	SP	Continuous	Site Pressure
S7	WD	Discrete	Wind Direction
S8	WS	Continuous	Wind Speed

This database is selected for our experiment because

1) It is taken from the real world
2) It is relatively large

3) It is of mixed-mode nature
4) All those parameters have their internal relationship based on the geographic location of the surface stations and might be governed by local terrain and land use
5) Some of the parameters are within geographical regions and some are meteorologically related.

Table 4.4.4 gives the list of parameters and a few examples of the data collected for each station.

We then applied ACA on this set of data. The Sum of Significant MR of the clustering process for various attribute cluster configurations are plotted on Table M-III. From the Sum of Significant MR values, it is obvious that a local optimal cluster configuration would consist of 5 clusters of parameters.

Table 4.4.4 MET Parameters and examples of their values

MET Surface Station A (Guangzhou)								MET Surface Station B (Shenzhen)							
A1	A2	A3	A4	A5	A6	A7	A8	B1	B2	B3	B4	B5	B6	B7	B8
J	J	10	9999	22	1021	23	3	A	A	11.2	9999	30	1024	17	5
J	J	9.6	9999	19	1021	24	3.8	A	A	10.8	9999	27	1024	6	4.5
J	J	9.3	9999	20	1021	20	3.4	A	A	10.2	9999	26	1024	9	3.6

MET Surface Station C (Dongguan)								MET Surface Station D (Dongguan)							
C1	C2	C3	C4	C5	C6	C7	C8	D1	D2	D3	D4	D5	D6	D7	D8
D	D	10.2	9999	35	1024	0	1.7	J	J	10.9	9999	31	1025	35	2.9
D	D	9.7	9999	32	1022	6	1.8	J	J	10.4	9999	30	1025	43	3
D	D	9.5	9999	29	1021	2	2.4	J	J	10.2	9999	31	1025	31	3.3

MET Surface Station E (Zhongshan)							
E1	E2	E3	E4	E5	E6	E7	E8
A	A	8.8	9999	28	1022	17	2.5
A	A	8.4	9999	28	1022	23	2.8

Table 4.4.5 ACA Run showing the value **k** for the local optimal cluster configuration.

No. of Attribute Cluster	**Sum of Significant MR**	**Number of Attribute Clusters**	**Sum of Significant MR**
K	**SMR**	***K***	**SMR**
2	11.42685035	**18**	9.23378
3	12.97939612	**19**	8.79266
4	13.44936898	**20**	6.62485
****5***	***13.65281846***	**21**	6.37775
6	11.98427219	**22**	6.58989
7	12.968216	**23**	6.05627
8	10.88940451	**24**	5.57822
9	11.50901881	**25**	4.22368
10	9.732935672	**26**	5.16222
11	11.8511256	**27**	3.99171
12	10.28084331	**28**	3.44625
13	12.0651708	**29**	1.83837
14	9.275470445	**30**	2.00887

15	7.77188342	**31**	1.8107
16	8.214801827	**32**	0.5296
17	9.642968239	**33**	1
18	9.23378089	**34**	0

* Highest Sum of Significant MR Implies Optimal *k* =5.

Table 4.4.6 show the grouping of the parameters in each of the parameter cluster.

The meaningful sub-grouping ---the attribute clusters obtained from ACA

By the highest value of SMR listed in Table 4.4.2, the mixed-mode meteorological database with 43 attributes has been clustered into 5 sub-groups. The first 4 of 5 clusters are grouped based on the interdependence among the similar characteristics (types) of the attributes within each cluster formed. This implies that those attributes within cluster are highly dependent upon each other or they are very “close” to each other or one “followed” by the others. We then study the mode and the characteristics of each of the clustered parameter groups.

Table 4.4.6 Attributes in the attribute clusters of the optimal cluster configuration

Attribute Group	Attribute Cluster Items
1 C	*B5, A5, C5, D5, E5 -- RH (Relative Humidity)
2 C	*C7, A7, B7, D7, E7 --WD (Wind Direction)
3 D	*C1, A1, B1, D1, E1 -- TC (Total Cloudiness)
4 C	*A6, B6, D6, E6, MM -- AP (Site Pressure)
5 M	*A3, A4, C6, B3, C3, D3, E3, A8, B8, C8, D8, E8, DD, HH (Dry Bulb Temperature & Wind Speed)

* The attribute marked with “*” is the mode of the attribute group.

The meaningfulness of the MODE attribute discovered for each cluster

One of our objectives of this study is to find out whether or not the mode discovered for each group can be considered as a meaningful governing attribute within the group.

This objective can be assessed by the following observation and analysis.

1) The reference parameter for the regional meteorological observation

For the first four clusters, the associated modes are the representative of each attribute type of the meteorological parameters in the region. In another words, the attribute selected by our algorithm as the MODE for each cluster is actually the most representative one within the cluster and thus be considered as the reference parameter among the set in those regions. That means that the reference parameters should have the most interdependence relations with others in the group. Thus the MODE for each group can be used as the reference parameter of the entire region.

2) The representative station for the regional meteorological observation

Within the five attributes (B5, C7, C1, A6, A3) being found as the MODE attributes for their respective clusters, we notice that attributes from stations of D and E are not there. This tells us that the two stations D and E are not very important for the weather observation in this area. From the practical operation view, if we just have adequate budget to operate two surface stations, we should set up A and C stations instead of B, D and E. Otherwise, if we just have good enough budget to operate two surface stations, we should set up A and C stations instead of B even D, E.

Observation of the role of regional parameters and the local parameters in the clusters

It has been observed that clusters 1,2,3,4 have the same type of the met parameters within respectively but cluster 5 has more than two types of the met parameters. From the clusters 1,2,3,4 we know that the attributes within each of them are regional meteorological parameters which have strong influence on each other in a large scale (tens or even hundreds kilometers). However, from cluster 5 we know that most of the attributes within this cluster are local met parameters and have less impact on each other in a small scale (a few kilometers).

By the P3 (Dry Bulb temperature) and P8 (Wind Speed) being fallen into one cluster, we could conjecture that they strongly influenced by the geographical factor LULC (land use and land coverage) or the surface roughness (geo-texture) which are not included in the database and thus not present in our clusters. It should be noted that the wind speed here is the surface wind speed instead of the up air wind speed. If the up air wind speed was collected and put into the database for clustering, it would likely induce a standalone cluster as the cluster 1,2,3,4 because the up air wind speed is also a typical regional parameter.

Summary: From the patterns discovered by our method, significant features within the data collected from the surface stations have been found which comply with the domain knowledge. Attributes in each of the first 4 clusters reflect the regional (global) characteristics of the correlated meteorological parameters. The mode found in each group has been treated as the reference parameters for those of the same type taken from the five stations. Regarding the last cluster group, all of the attributes therein reflect local characteristics significantly influenced by the local geographical feature such as land use and land coverage. The discovered modes in these clusters cover only 3 stations indicating that the remaining two are in very weak position for the weather condition analysis.

4.5. Experiment on Delay Coking Database

This is a very large set of data. The data is taken from the delay coking unit (DCU) of the Sinopec SJZ Petro-Chemical refinery for about 5-month-long period. It was acquired directly from the ABB DCS sensors by which the temperatures, the levels, the flow rates and the pressures as well as the control actions of PLCs were collected.

Delayed coking is a semi-continuous thermal cracking process in which a heavy hydrocarbon feedstock is converted to lighter and more valuable products and coke. The mechanism of coking can be broken down to ***three*** distinct stages.

The feed undergoes partial vaporization and mild cracking as it passes through a specially designed coking furnace.

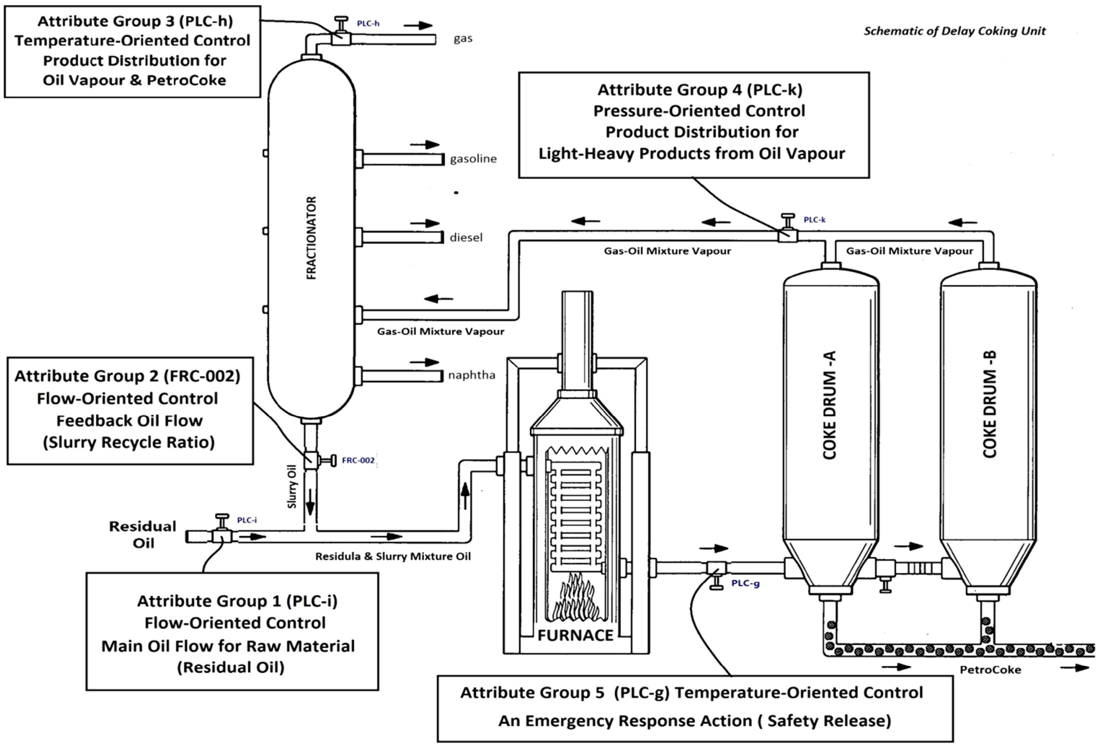

Attribute Group 3 (PLC-h)
Temperature-Oriented Control
Product Distribution for
Oil Vapour & PetroCoke
PLC-h
gas
Schematic of Delay Coking Unit
Attribute Group 4 (PLC-k)
Pressure-Oriented Control
Product Distribution for
Light-Heavy Products from Oil Vapour
gasoline
FRACTIONATOR
diesel
PLC-k
Gas-Oil Mixture Vapour
Gas-Oil Mixture Vapour
Gas-Oil Mixture Vapour
naphtha
Attribute Group 2 (FRC-002)
Flow-Oriented Control
Feedback Oil Flow
(Slurry Recycle Ratio)
FRC-002
Slurry Oil
COKE DRUM -A
COKE DRUM -B
Residual
Oil
PLC-i
Residula & Slurry Mixture Oil
PLC-g
FURNACE
Attribute Group 1 (PLC-i)
Flow-Oriented Control
Main Oil Flow for Raw Material
(Residual Oil)
PetroCoke
Attribute Group 5 (PLC-g) Temperature-Oriented Control
An Emergency Response Action (Safety Release)

Figure 4.5.. The Schematic of Delay Coking Unit.

The vapours undergo cracking as they pass through the coke drum to fractionation facilities downstream where products of gas, naphtha, jet fuel and gas oil are separated. The petroleum coke remains in the drum.

The heavy hydrocarbon liquid trapped in the coke drum is subjected to successive cracking and polymerization until it is converted to vapour and coke.

The residuum (fresh feed) from the Hydrocracker Fractionation Unit enters the bottom section of the fractionator where material lighter than the desired cut point of the coke gas oil is flashed off and the remaining material combines with the recycle material condensed in the bottom of the fractionator to form the combined feed.

This combined feed is then routed to the charge furnace where the liquid is heated to its incipient coking temperature to produce vapourization and mild cracking. Steam is injected into the furnace feed line to prevent coke deposition in the furnace coils, increase tube velocity and reduce hydrocarbon partial pressure.

The vapour/liquid mixture then enters the bottom of the coke drum where the vapour experiences further cracking and the liquid experiences successive cracking and polymerization until it is completely converted to vapour and coke. The coke drum effluent vapour enters the fractionator where the hot vapour are quenched with wash oil. The condensed portion then forms the recycle stream and is recycled to the furnace for another pass through the coke drum. The condensed vapour is fractionated into gas, naphtha, jet fuel and gas oil. Gas oil and jet fuel are removed as side cuts and routed to the Gas Oil

Hydrotreater and the Naphtha/Jet Hydrotreater.

Typically, a DCU could consist of three main processing sections: the heating units (furnace), the coking drums and the fractionator (tower) (Figure 4.5.1).

The raw fresh material flow (also called residual oil) has been filled into and then be heated by the heating unit (by which the flow rate of the residual oil must be controlled carefully to keep flowing to avoid the residual oil becoming coke inside the heated pipe lines, blocking the heating unit) and then be pumped into the following delay coking units (Coke Drum) to produce the coke.

For the two coking units, the pressure has been carefully controlled to avoid the conversion of the mixed oil-gas flow into coke totally and more oil vapour will be expected to get out from the top of the drums for producing more “light” products. At the same time, the mixed oil-gas vapour flow arises at the top of a delay coking unit will be introduced into the fractionator tower to produce the different oil products like gasoline, diesel, naphtha and etc. according their different “cutting” temperatures respectively, inside which the temperature should be carefully controlled to “cut out” the product distribution expected for different market purposes.

Table 4.5.1 gives a brief description of the Delay Coking Database. It consists of 22,096 samples and 47 attributes out of which 11 of them are discrete valued data and 36 are continuous valued data. Since this is a set of very complex data taken directly from the delay cooking plant, there is no specific class information. It is relatively a large database. Since we have a certain degree of partial domain knowledge concerning this system, this set of data will be ideal to challenge the usefulness and effectiveness of the proposed system.

Table 4.5.1 Data Description of the Large Database obtained from a Delay Coking Unit

Data Description				
Data Set	Attribute Characteristics	No. of Samples	No. of Attributes	No. of Classes
Coking Data	Mixed-Mode Data	22096	47 D11 & C36	Unknown

We first apply ACA to cluster the database into sub-database containing subgroups of attributes. Table C-II show the Sum of the Significant MR values for different attribute cluster configurations. It is found that k=5 would render a local optimal configuration. Figure 4.5.2 gives the k-SSMR plots taken from our ACA Algorithm.

We next proceed to discretize the continuous data for each cluster based on the mode discovered for that cluster. We then display the results of each cluster and conduct the in-depth analysis to derive the meaning from the patterns and rules discovered for each set of mixed mode sub-database. For the sake of exposition, we refer each cluster formed by the feature characteristics of the cluster. Here, we shall proceed with the in-depth analysis of each attribute cluster in the light of the partial domain knowledge available.

The Cluster Group 1 (Table 4.5.4) associated with flow-oriented control of main oil flow for Raw Materials (Residual Oil) with its governing attribute PLC-i has supportive evidence from the system characteristics of the operating plant that the attribute PLC-i is actually acting as the global control factor for the entire processing system. In reality, the Cluster Group 1 containing parameters pertaining to a flow-oriented control group in which the flow controller PLC-i plays a definitely role in governing the values of all of the other parameters within this group. This means that any changes from the MODE attribute PLC-i will influence the settings or readings of the others within this group whose members distributes almost everywhere in the entire coking plant.

The Cluster Group 2 associated with the flow-oriented control of the feedback oil flow (also known as slurry recycle ratio) with its governing attribute FRC-002 turns out to be a very important subsystem for delay coking system. This subsystem, referred to as the Slurry Recycle subsystem, is the subsystem by which a significant parameter (Slurry Recycle Ratio) has been used as a decision and monitoring factor. Actually, the Cluster Group 2 represents a local flow-oriented control group for a coking system by which the Recycle Ratio is working closely together with other elements within the group. If the operation director wishes to shift the coking facility to work on another recycle ratio, he will adjust the governing attribute FRC-002, monitor the other parameters in the subsequent operations to ensure that the current adjustment will complete successfully. Thus any changes from the MODE attribute FRC-002 will locally influence the settings or readings of the other parameters within this local group of which the components are installed around the feedback pipe system.

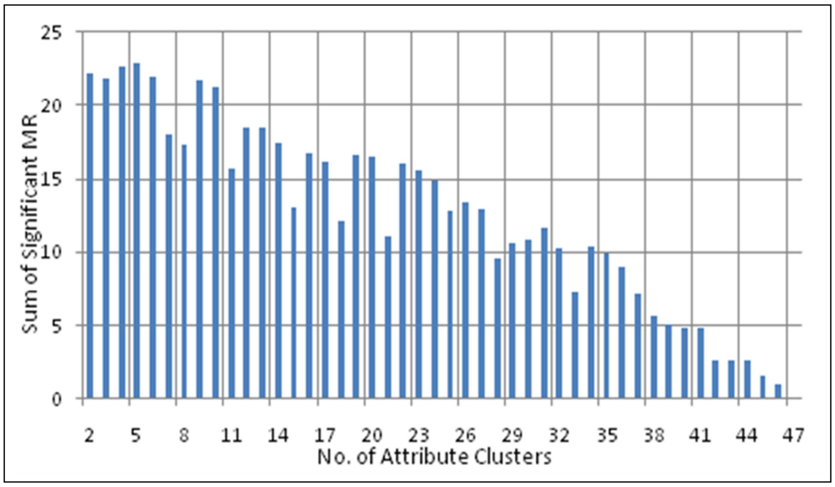
Cluster Group 3 (Table 4.5.5) associated with temperature-oriented control of the product distribution for oil vapor and petro-coke with its governing attribute PLC-h represents another very important subsystem for the delay coking plant. This system can be considered as the Fractionator subsystem by which the oil vapor flow has been guided into the tower and produces different products based on the required product distribution. Actually, the Cluster Group 3 is a local temperature-oriented control group for the coking system by which the expected final product distribution has been made by carefully building up the “cutting temperature” of the Fractionator through the controller PLC-h. The values of the other parameters will follow up the changing of the PLC-h reading to reach their new values.

Table 4.5.2 Sum of the Significant MR values for different attribute cluster configurations.

No. of Attribute Cluster	**Sum of Significant MR**	**No. of Attribute Cluster**	**Sum of Significant MR**
K	**SMR**	***K***	**SMR**
2	22.2426	**24**	14.8637
3	21.8947	**25**	12.7643
4	22.628	**26**	13.4253
***5**	22.9081	**27**	12.9122
6	21.9604	**28**	9.55878
7	17.9875	**29**	10.5888
8	17.3895	**30**	10.8342
9	21.7116	**31**	11.7177
10	21.3002	**32**	10.2624
11	15.7158	**33**	7.24438

12	18.5434	**34**	10.3341
13	18.4806	**35**	9.95252
14	17.395	**36**	9.03594
15	13.0992	**37**	7.10925
16	16.7974	**38**	5.63904
17	16.2262	**39**	5.09977
18	12.1681	**40**	4.83713
19	16.6476	**41**	4.84365
20	16.5275	**42**	2.59895
21	11.0865	**43**	2.63305
22	16.0462	**44**	2.58523
23	15.5782	**45**	1.53357
24	14.8637	**46**	0.95916
25	12.7643	**47**	0

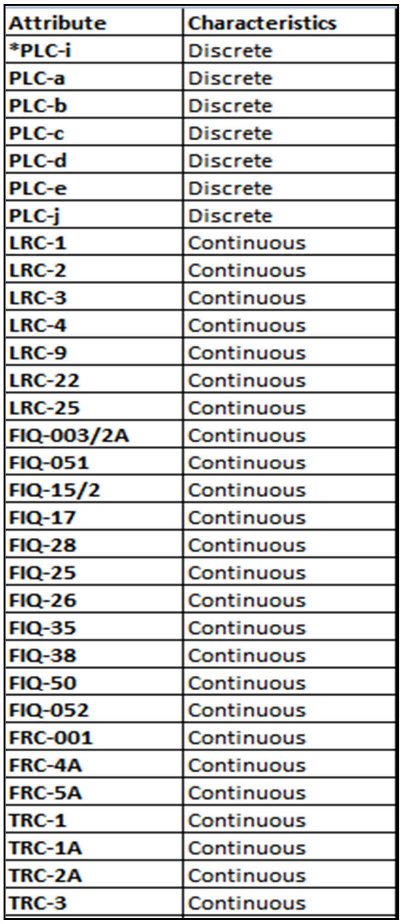
* Highest Sum of Significant MR Implies Optimal *k* =5.



* Highest Sum of Significant MR Implies Optimal *k* =5.

Figure 4.5.2 Plot of Sum of Significant MR values against k, the number of attribute clusters.

Table 4.5.3 Cluster 1: Flow-Oriented Control of Main Oil Flow for Raw Materials (Residual Oil).



Attribute	Characteristics
*PLC-i	Discrete
PLC-a	Discrete
PLC-b	Discrete
PLC-c	Discrete
PLC-d	Discrete
PLC-e	Discrete
PLC-j	Discrete
LRC-1	Continuous
LRC-2	Continuous
LRC-3	Continuous
LRC-4	Continuous
LRC-9	Continuous
LRC-22	Continuous
LRC-25	Continuous
FIQ-003/2A	Continuous
FIQ-051	Continuous
FIQ-15/2	Continuous
FIQ-17	Continuous
FIQ-28	Continuous
FIQ-25	Continuous
FIQ-26	Continuous
FIQ-35	Continuous
FIQ-38	Continuous
FIQ-50	Continuous
FIQ-052	Continuous
FRC-001	Continuous
FRC-4A	Continuous
FRC-5A	Continuous
TRC-1	Continuous
TRC-1A	Continuous
TRC-2A	Continuous
TRC-3	Continuous

Here, the attribute PLC-i marked with “*” is the mode of the attribute group. A mode is with the highest normalized mutual information in the attribute group.

Table 4.5.4 Cluster 2: Flow Oriented Control Feedback Oil Flow Recycle Ratio)

Attribute	**Characteristics**
*** FRC-002**	Discrete
LRC-5	Continuous
FIQ-004	Continuous
FIQ-20	Continuous
FIQ-22	Continuous

* The attribute FRC-002 marked with "*" is the mode of the attribute group. A mode is with the highest normalized mutual information in the attribute group.

Table 4.5.5 Cluster 3: Temperature-Oriented Control of Production Distribution for Oil Vapor and petro-coke.

Attribute	**Characteristics**
*** PLC-h**	Discrete
PLC-f	Discrete
TR-15A-17	Continuous
TR-15A-19	Continuous

* The attribute PLC-h marked with "*" is the mode of the attribute group. A mode is

with the highest normalized mutual information in the attribute group.

The Cluster Group 4 (Table 4.5.6) is associated with pressure-oriented control of production distribution for light-heavy products from oil vapor. Its governing attribute PLC-k is a very important parameter in the Coke Drum of the delay coking system. The heated residual flow is filled into the drums and it will mainly be divided into two parts, one is petro-coke and the other is oil vapor. Actually, the Cluster Group 4 represents a local pressure-oriented control group for a coking system by which the setting of the temperature will determine the coke production ratio or distribution. Thus, the MODE attribute PLC-k in this cluster will locally influence the coke-vapor distribution of the drums.

Table 4.5.6 Cluster 4: Pressure-Oriented Control of Production Distribution for Light-Heavy Products from Oil Vapor.

Attribute	**Characteristics**
*** PLC-k**	Discrete
FIQ-21	Continuous
PRC-8	Continuous

* The attribute PLC-k marked with "*" is the mode of the attribute group. A mode is with the highest normalized mutual information in the attribute group.

Table 4.5.7 Cluster 5: Temperature-Oriented Control of Emergency Response Action (Safety Release).

Attribute	**Characteristics**
*** PLC-g**	Discrete
TR-15A-18	Continuous

* The attribute PLC-g marked with "*" is the mode of the attribute group. A mode is with the highest normalized mutual information in the attribute group.

The Cluster Group 5 (Table 4.5.7) is associated with temperature-oriented control for emergency response actions. Its governing attribute PLC-g is acted as an emergency control unit which responds to the overheated condition of the heating unit (furnace). It helps to ensure that the entire delay coking system would work under a safety situation. Its major function is to control the heating unit.

Summary: Based on the five clusters from our developed method for the patterns, the most important relationships with the sensors and controllers of the coking facilities have been found: including the temperature-oriented groups, pressure-oriented groups and flow-oriented groups. The attribute number and distribution of the largest group indicates that its mode acts as a control factor for the entire processing system and has globally influenced almost all of the process parameters for the facility.

From the parameter grouping, the discovered results indicate that the other two groups control the output distributions of the two internal units like coke drum and fractionators. They are very important groups for the local performances of the processing usually referred to as performance factor.

The last group discovered is exactly associated with the critical safety mechanism designed for this pressure-temperature-mixed processing facility. Its mode is actually controls the temperature condition as a trigging factor to activate the emergency release response.
All of the five cluster groups with the patterns and mode attributes discovered provided us the stronger analysis evidence for the whole industry system's control principal.

Chapter 5

Conclusion and Future Research

The research presented in this dissertation was motivated by the challenges we are confronting today: (1) an increasingly huge amount of raw mixed-mode data today require effective pattern discovery methods to unveil inherent subtle information for better understanding; (2) the pressing need to develop intelligent systems which are able to support knowledge discovery and decision support from overwhelming volume of discovered patterns; (3) the increasing demand of applications of discovered patterns in scientific, business and industry; and (4) the application limitation of most existing systems which are not general enough to solve problems on mixed-mode databases with numerous real-world applications.

The research works presented in this thesis have provided an integrated, flexible and generic framework for pattern discovery and analysis of large mixed-mode databases. Its applications cover databases with continuous, categorical and mixed- mode data. Based on the well defined problems and research objectives stated in Chapter 1, the developed research methods presented in Chapter 3, and the broad applications on real world and industrial problems presented in Chapter 4, the contribution of the thesis research in theoretical and methodological perspectives as well as in real world applications have been conveyed. The validity and the effectiveness of the proposed methods has been backed by a number of successful experimental results. Their usefulness in real world applications has been demonstrated by the intriguing and revealing results obtained when applying to two large mixed-mode databases --- one consists of a large set of meteorological data taken from a geographic area in Southern China and another is a set of massive multi-senor data taken from a delay coking plant.

5.1 Summary of Contributions

5.1.1 Theoretical Contributions

With the defined research work and proposed methods completed, the theoretical contribution can be outlined as below.

1) Development of a theoretical framework for pattern discovery for mixed-mode data at event level.

A theoretical framework has been developed for discovery of high order patterns for mixed-mode data (which include continuous data, categorical data and a combination of the two) at event level. By converting continuous data into interval events under a general problem setting, it shifts the basic data representation units into events. It thus provides a unified framework to define association patterns as event associations. It thus generalizes the pattern discovery and data mining methodologies to cover the important mixed-mode data under a unified event based framework. Allowing probabilistic variations and statistical justification, it brings forth a unified system for pattern discovery, data mining and machine learning. The experimental results obtained show that once the patterns are organized at the event level, they can be interpreted and understood much more easily. A unique characteristic of this theoretical framework is its natural accommodation of local organization of event associations in various event subspaces of lower dimension.

2) Demonstration of the necessity of attributes clustering in large databases and the provision of an attribute clustering algorithm for mixed-mode data.

From the experimental results on certain UCI data sets as well as on the two large databases of the real world problems, the thesis furnishes significant evidences that strong correlation attribute groups exist in large databases and their discovery might shed light to the how features are associated within the databases and how the discovered association patterns may impact class definition and the attribute group interactive activities. The contribution of this thesis is not only stating the problem but providing an algorithmic solution to partition the databases accordingly. It also reveals the feature relationship and association characteristics of each of the clustered groups.

5.1.2 Methodological Contributions

1) **The provision of algorithmic procedure to obtain normalized mutual information between mixed-mode attributes.**

   For mixed-mode data, one of the major hurdle in assessing interdependence between heterogeneous attributes (i.e. between discrete and discrete, discrete and continuous and between continuous and continuous attributes) is the lack of a implementable measure to account for the interdependence between attributes of mixed types. In this dissertation, the normalized mutual information between these three pair of attribute types have been defined, implemented and tested over large sets of data. They have been used in the finding of the mode, the governing attributes, the intrinsic class attributes and in the k-mode attribute clustering algorithm.

2) **Discovery of mode and governing attributes for a mixed-mode data set.**

   With the normalized mutual information computed between all attribute pairs for an attribute set, the mode can be obtained as the attribute with the highest sum of normalized statistical significant mutual information with all other attributes in a mixed-mode data set. The identified mode has been used in the k-mode attribute clustering algorithm as well as in driving the discretization of continuous data in the data set.

3) **Discretization of continuous data in a mixed-mode data set.**

   One of the major impediments blocking the application of pattern discovery for mixed-mode data is that there has been no easy way prior to this dissertation for discretizing the continuous data in a database setting when class information is absent or unavailable. The contribution of this thesis is that normalized mutual information measures between different types of attributes have been implemented for two separate stages of the pattern analysis --- the attribute clustering phase and the continuous data discretization phase.

   In solving the discretization problems, two issues have been raised and later justified by enormous experimental evidences. The first is the idea of the possible existence of a governing or most representative attributes. One may refer it as an intrinsic class attribute or a governing attribute for a correlated data set. Such an attribute, if found and justified, can be used to drive the discretization of the continuous attributes. However, how strong this attribute depends on the strength of its summed interdependencies with others in the attribute group. Once a reasonable one is identified it could be used to drive the discretization of the continuous data in the group just like the class attribute does. The second is related to the necessity of attribute clustering. For a very large database, unless a class label is given or assumed, with the absence of class information, there is no reason to believe that the entire database is governed by a single attribute. There could be several correlated attribute groups existing inherently in the data set. Each may share more correlated information among themselves than with other groups. Thus it is not meaningful to use the mode of a large data set to drive the discretization. A more reasonable approach is that we should first find out whether the database could be optimally partitioned into several coherent attribute groups before discretization be applied to each group like we have observed in the application on the colon cancer data. Once found, we could apply discretization of continuous data to each attribute group. A contribution of thesis is that it has provided evidences to demonstrate this happened and the proposed solutions work.

5.1.3 Application Contributions

1) **Automatic grouping, repooling and discretization of gene expressions for analyzing and classifying genes without relying on class information.**

That the proposed methods are able to show that both the gene clustering and the re-pooling process work for continuous gene expression data as effectively as in the cases when class information is provided represents a huge advancement of gene expression analysis. This capability not only speeds up the diagnostic process but also reveal the gene interactive patterns for various types of gene tissues at various histological or pathological stages objectively. That the use of the discretized gene expression intervals to achieve high classification rate of cancer and normal cells equivalent to systems using class labels implies that not only the concept of governing attribute works for discretization but also could be used to reveal the interactive role of the governing genes with others. .

2) Discovery and grouping of meteorological patterns from surface stations over a large area rendering subtle information for regional weather monitoring.

The discovery and grouping of meteorological measurement patterns from data taken from various surface stations in a wide area reflect the regional and global characteristics of the correlated meteorological parameters. The consistency and the representative characteristics of each of the meteorological modes discovered suggest that certain modes could serve as reference parameters as they renders much more precise assessment of the weather monitoring system. Other subtle patterns may reveal the impact of land use and land coverage. Its significance requires further analysis.

3) The discovery and grouping of parameter patterns in delay coking process revealing system function and operational characteristics.

The pattern discovery and grouping experiment on a large set of sensed and control data set taken from a delay coking plant yields most important relationships among sensors and controllers of the coking facilities. From the attribute number and distribution of the largest correlated group, the most significant control factor which has global influence over almost all of the process parameters in the facility is located and its interactive patterns with others have been discovered. From the parameter grouping, the discovered results indicate that the other two groups control the output distributions of the two internal units like coke drum and fractionators. It is surprising to find that a two parameter group discovered is associated exactly with the critical safety mechanism designed for this pressure-temperature-mixed processing facility. Its mode is actually controls the temperature condition and serves as a trigging factor to activate the emergency release response. Such findings show the usefulness and effectiveness of the proposed method in revealing subtle operation patterns for system monitoring, control and optimization.

In summary, the results of the dissertation research open the door for more precise system behavior analysis and modeling. It is fulfilling the vision that: “ through pattern discovery on large mixed-mode databases, we are one step closer to meeting the challenge: “ *from data to model to knowledge*” in the petabyte age”.

5.2 Suggested Future Research

This dissertation has developed a basic framework for discovering patterns for mixed-mode data. It is expected that there will be considerable refinement of the system to arrive at an integrated prototype for researchers and general users. Here we will list some of the suggested future research.

1) Refining the pattern discovery framework for large mixed-mode databases will be continued.
2) Special attention will be devoted to explore the characteristics of the governing attributes including exploring of its patterns and pattern clusters with other attributes.
3) With the class labels removed and discretization problem solved, the technology developed for pattern clustering on categorical data only can now be applied to continuous and mixed-mode data. Thus, a natural extension of this research is to integrate the system with pattern clustering, summarization and visualization for mixed-mode data.
4) Since the proposed system is able to produce insightful patterns and solutions to two of the difficult real world problems with large databases, extensive effort to apply this new technology to other large mixed-mode data is planned. By relating both the subgroup and the entire group patterns to the application domain, means to generate models and knowledge will be explored.
5) Development of an integrated system for pattern discovery, pattern clustering, summarization and visualization system for mixed-mode data with or without class information --- a worth achieving goal of pattern discovery.

References

[1] A. K. C. Wong and G. C. L. Li, “Association Pattern Analysis for Pattern Pruning, Pattern Clustering and Summarization”, to appear in Journal of Knowledge and Information Systems, 2010.

[2] AKC Wong and G Li, “Simultaneous Pattern Clustering and Data Grouping”, IEEE Trans Knowledge and Data Engineering, Vol. 20, No. 7, pp 911-923, 2008.

[3] G Li and AKC Wong, “Pattern Distance Measures in Categorical Data for Pattern Pruning and Clustering”, submitted to IEEE Trans. on Knowledge and Data Engineering.

[4] L Liu, AKC Wong, and Y Wang, “A Global Optimal Algorithm for Class-Dependent Discretization of Continuous Data”, Intelligent Data Analysis, Vol 8, no 2, pp 151-170, 2004

[5] AKC Wong and Y Wang, ‘Pattern Discovery: A Data Driven Approach to Decision Support,’ IEEE SMC, Vol 33, no. 3, pp. 114-124, 2003.

[6] Y Wang and AKC Wong, “From Association to Classification: Inference Using Weight of Evidence,’ IEEE Trans On Knowledge Systems, Vol 15, no 3, pp 914-925, 2003

[7] T Chau and AKC Wong, ‘Pattern Discovery by Residual Analysis and Recursive Partitioning,’ IEEE Trans on Knowledge and Data Engineering, pp 833-854, 1999

[8] AKC Wong, and Y Wang, ‘High-Order Pattern Discovery from Discrete-Valued Data,’ IEEE Trans On Knowledge Systems, pp 877-893,Vol 9, No 6, 1997

[9] JY Ching, AKC Wong, and Chan, KCC, “Class-dependent Discretization for Inductive Learning from Continuous and Mixed-Mode Data”, IEEE PAMI, Vol 17, No 7, pp 641-651, July 1995

[10] K K Durston, D KY Chiu, A KC Wong and G CL Li, “Inferring higher-order structure in protein sequences: a granular computing analysis” submitted to BMC Genomics.

[11] AKC Wong, WH Au and KCC Chan, "Discovering High-Order Patterns of Gene Expression Levels", Journal of Computational Biology, Vol. 15, No.6, 2008. revision, 2008

[12] WH Au, KCC Chan, AKC Wong and Y Wang, “Attribute Clustering for Grouping, Selection, and Classification of Gene Expression Data” IEEE/ACM Trans on Computational Biology and Bioinformatics, Vol 2, No2, pp 83-101, 2005.

[13] A.K.C. Wong, Information Pattern Analysis, Synthesis and Discovery, Chapter 7, pages 254–257. University of Waterloo, 1998.

[14] A. K. C. Wong and Y. Wang, “High Order Pattern Discovery from Discrete-Valued Data,” IEEE Trans. on Knowledge and Data Eng., vol. 9, no. 6, pp. 877-893, 1997.

[15] A.K.C. Wong and Y. Wang, “Pattern Discovery: A Data Driven Approach to Decision Support,” IEEE Trans. on Syst., Man, Cybern. – Part C, vol. 33, no. 1, pp. 114-124, 2003.

[16] A.K.C. Wong, D.K.Y. Chiu and W. Huang, ‘A Discrete-Valued Clustering Algorithm with Applications to Bimolecular Data,’ Information Sciences, vol. 139, pp. 97-112, 2002.

[17] A. K. C. Wong and D. K. Y. Chiu, “Synthesizing Statistical Knowledge from Incomplete Mixed-Mode Data,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 9, no. 8, pp. 796-805, 1987

[18] A.K.C. Wong and C.C. Wang. DECA - a discrete-valued data clustering algorithm. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1(4):342–349, 1979. 27

[19] A.K.C. Wong and T.S. Liu, ‘Typicality, Diversity and Feature Patterns of an Ensemble,’ IEEE Trans. on Computers, vol. 24, no. 2, pp. 158-181, 1975

[20] A.K.C. Wong, T.S. Liu, and C.C. Wang, “Statistical Analysis of Residue Variability in Cytochrome C,” Journal of Molecular Biology, vol. 102, pp. 287-295, 1976.

[21] Wai-Ho Au, Keith C.C. Chan, Andrew K.C. Wong, and Yang Wang, Attribute Clustering for Grouping, Selection, and Classification of Gene Expression Data, IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, VOL. 2, NO. 2, APRIL-JUNE 2005".

[22] C. C. Wang and A. K. C. Wong, "Classification of Discrete-Valued Data with Feature Space Transformation," IEEE Trans. on Automatic Control, vol. AC-24, no. 3, pp. 434–437, 1979.

[23] D. Chiu, A. Wong, and B. Cheung. Information discovery through hierarchical maximum entropy. Journal of Experimental and Theoretical Artificial Intelligence, 2:117–129, 1990.

[24] J. Catlett. On changing continuous attributes into ordered discrete attributes. In Y. Kodratooe, editor, Proc. 5th European Working Session on Learning, pages 164–178, Porto, Portugal, 1991, March. Springer-Verlag Heidelberg.

[25] J. Y. Ching, A. K. C. Wong, and K. C. C. Chan, "Class-Dependent Discretization for Inductive Learning from Continuous and Mixed-Mode Data," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 17, no. 7, pp. 631-641, 1995.

[26] P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In Proc. Fifth European Working Session on Learning, pages 151–163, Berlin, 1991. Springer.

[27] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In International Conference on Machine Learning, pages 194–202, San Francisco, CA, 1995.

[28] Usama M. Fayyad and Keki B. Irani. On the handling of continuous-valued attributes in decision tree generation. Machine Learning, 8(1):87–102, 1992.

[29] K. M. Ho and P. D. Scott. Zeta: A global method for discretization of continuous variables. In D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors, Knowledge Discovery and Data Mining, pages 191–194, Menlo Park, 1997. AAAI Press.

[30] Robert C. Holte. Very simple classification rules perform well on most commonly used datasets. Machine Learning, 11(1):63–90, 1993.

[31] Quinlan J.R. Induction of Decision Trees. Machine Leariining 1, 1986.

[32] Quinlan J.R. C4.5:programs for Machine Learning. Morgan Kaufmann, 1993.[12] Tou J.T and Gonzalez R.C. Pattern Recognition Principles. Addison-Wesley, 1974.

[33] J.Y.Ching, A.K.C.Wong, and K.C.C.Chan. Class-dependent discretization for inductive learning from continuous data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(7):641–651, 1995.

[34] R. Kerber. Chi merge: Discretization of numeric attributes. In Proceedings of the 9th International Conference on Artificial Intelligence, pages 123–128, Menlo Park CA, 1992.

[35] R. Kohavi, G. John, R. Long, D. Manley, and K. Pfleger. Mlc++: A machine learning library in c, 1994.

[36] T Kohonen. Self-Organization and Associative Memory. Springer-Verlag, Berlin, Germany, 1989.

[37] L. Kurgan and K.J. Cios. Discretization algorithm that uses class-attribute interdependence maximization. In Proceedings of the 2001 International Conference on Artificial Intelligence (IC-AI 2001), pages 980–987, Las Vegas, Nevada, 2001,JUNE.

[38] P. Langley. Induction of recursive bayesian classifiers. In P. Brazdil, editor, ECML93, volume 667 of LNAI, pages 153–164, Berlin, 1993. SV.

[39] Pat Langley, Wayne Iba, and Kevin Thompson. An analysis of bayesian classifiers. In National Conference on Artificial Intelligence, pages 223–228, San Jose, California, 1992.

[40] Huan Liu and Rudy Setiono. Feature selection via discretization. IEEE Transactions on Knowledge and Data Engineering, 9(4):642–645, 1997.

[41] P.M. Murphy and D.W. Aha. Uci repository of machine learning databases, 1994.

[42] A. Paterson and T.B. Niblett. Acls manual. Technical report, Intelligent Terminals Ltd., Edinburg, 1987.

[43] Bernhard Pfahringer. Compression-based discretization of continuous attributes. In International Conference on Machine Learning, pages 456–463, San Francisco, CA, 1995.

[44] J. R. Quinlan. Simplifying decision trees. International Journal of Man-Machine Studies, 27(3):221–234, 1987.

[45] J.R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kauffman, San, Mateo CA, 1993.

[46] M. RIcheldi and M. Rossotto. Class-driven statistical discretization of continuous attributes (extended abstract). Springer-Verlag, Berlin, Heidelberg, 1995.

[47] Moshe Sniedovich. Dynamic Programming, chapter appendix, pages 348–350. New York, 1992.

[48] R. Agrawal, S. Ghost, T. Imielinski, B. Iyer, and A. Swami, “An Interval Classifier for Database Mining Applications,” in Proc. of the 18th Int’l Conf. on Very Large Data Bases, Vancouver, British Columbia, Canada, 1992, pp. 560–573.

[49] R. Agrawal, T. Imielinski, and A. Swami, “Mining Association Rules between Sets of Items in Large Databases,” in Proc. of the ACM SIGMOD Int’l Conf. on Management of Data, Washington D.C., 1993, pp. 207–216.

[50] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules,” in Proc. of the 20th Int’l Conf. on Very Large Data Bases, Santiago, Chile, 1994, pp. 487–499.

[51] Stephen D. Bay. Multivariate discretization for set mining. Knowledge and Information Systems, 3(4):491–512, 2001.

[52] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, “Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays,” Proc. of the National Academy of Sciences of the United States of America, vol. 96, no. 12, pp. 6745–6750, 1999.

[53] W.H. Au, K.C.C. Chan, A.K.C. Wong and Y. Wang, “Attribute Clustering for Grouping, Selection and Classification of Gene Expression Data,” to appear in IEEE Trans. on Computational Biology and Bioinformatics, 2005.

[54] W.H. Au and K. C. C. Chan, “Classification with Degree of Membership: A Fuzzy Approach,” in Proc. of the 1st IEEE Int’l Conf. on Data Mining, San Jose, CA, 2001, pp. 35–42.

[55] W.H. Au and K. C. C. Chan, “Mining Fuzzy Association Rules in a Bank-Account Database,” IEEE Trans. on Fuzzy Systems, vol. 11, no. 2, pp. 238–248, 2003.

[56] W.H. Au, K. C. C. Chan, and X. Yao, “A Novel Evolutionary Data Mining Algorithm with Applications to Churn Prediction,” IEEE Trans. on Evolutionary Computation, vol. 7, no. 6, pp. 532–545, 2003.

[57] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, “Tissue Classification with Gene Expression Profiles,” in Proc. of the 4th Annual Int’l Conf. on Computational Molecular Biology, Tokyo, Japan, 2000.

[58] K. C. C. Chan and W.H. Au, “Mining Fuzzy Association Rules in a Database Containing Relational and Transactional Data,” in A. Kandel, M. Last, and H. Bunke (Eds.), Data Mining and Computational Intelligence, New York, NY: Physica-Verlag, 2001, pp. 95–114.

[59] Y. Cheng and G. M. Church, “Biclustering of Expression Data,” in Proc. of the 8th Int’l Conf. on Intelligent Systems for Molecular Biology, San Diego, CA, 2000, pp. 93-103.

[60] D. K. Y. Chiu and A. K. C. Wong, “Multiple Pattern Associations for Interpreting Structural and Functional Characteristics of Biomolecules,” Information Sciences, vol. 167, pp. 23–39, 2004.

[61] M. Delgado, N. Márin, D. Sánchez, and M.-A. Vila, “Fuzzy Association Rules: General Model and Applications,” IEEE Trans. on Fuzzy Systems, vol. 11, no. 2, pp. 214–225, 2003.

[62] F. De Smet, J. Mathys, K. Marchal, G. Thijs, B. De Moor, and Y. Moreau, “Adaptive Quality-Based Clustering of Gene Expression Profiles,” Bioinformatics, vol. 18, no. 5, pp. 735–746, 2002.

[63] C. Ding and H. Peng, “Minimum Redundancy Feature Selection from Microarray Gene Expression Data,” in Proc. of the IEEE Computational Systems Bioinformatics Conf., Stanford, CA, 2003, pp. 523–528.

[64] E. Domany, “Cluster Analysis of Gene Expression Data,” Journal of Statistical Physics, vol. 110, pp. 1117–1139, 2003.

[65] S. Dudoit, J. Fridlyand, and T. P. Speed, “Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data,” Journal of the American Statistical Association, vol. 97, no. 457, pp. 77–87, 2002.

[66] N. Friedman, M. Nachman, and D. Pe’er, “Using Baysian Networks to Analyze Expression Data,” in Proc. of the 4th Annual Int’l Conf. on Computational Molecular Biology, Tokyo, Japan, 2000, pp. 127–135.

[67] K. Hirota and W. Pedrycz, “Fuzzy Computing for Data Mining,” Proc. of the IEEE, vol. 87, no. 9, pp. 1575–1600, 1999.

[68] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data Clustering: A Review,” ACM Computing Surveys, vol. 31, no. 3, pp. 264–323, 1999.

[69] D. Jiang, C. Tang, and A. Zhang, “Cluster Analysis for Gene Expression Data: A Survey,” IEEE Trans. on Knowledge and Data Engineering, vol. 16, no. 11, pp. 1370–1386, 2004.

[70] J. Kacprzyk and S. Zadrozny, “On Linguistic Approaches in Flexible Querying and Mining of Association Rules,” in H. L. Larsen, J. Kacprzyk, S. Zadrozny, T. Andreasen, and H. Christiansen (Eds.), Flexible Query Answering Systems: Recent Advances, Proc. of the 4th Int’l Conf. on Flexible Query Answering Systems, Heidelberg, Germany: Physica-Verlag, 2001, pp. 475–484.

[71] A. D. Keller, M. Schummer, L. Hood, and W. L. Ruzzo, “Bayesian Classification of DNA Array Expression Data,” Technical Report UW-CSE-2000-08-01, Department of Computer Science and Engineering, University of Washington, 2000.

[72] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer, “Classification and Diagnostic Prediction of Cancers Using Gene Expression Profiling and Artificial Neural Networks,” Nature Medicine, vol. 7, no. 6, pp. 673–679, 2001.

[73] T. Kohonen, Self-Organizing Maps, 3rd Ed., Berlin, Germany: Springer-Verlag, 2001.

[74] J. Li and L. Wong, “Identifying Good Diagnostic Gene Groups from Gene Expression Profiles Using the Concept of Emerging Patterns,” Bioinformatics, vol. 18, no. 5, pp. 725–734, 2002.

[75] B. Liu, W. Hsu, and Y. Ma, “Integrating Classification and Association Rule Mining,” in Proc. of the 4th Int’l Conf. on Knowledge Discovery and Data Mining, New York, NY, 1998, pp. 80–86.

[76] L. Liu, A. K. C. Wong, and Y. Wang, “A Global Optimal Algorithm for Class-Dependent Discretization of Continuous Data,” Intelligent Data Analysis, vol. 8, no. 2, pp. 151–170, 2004.

[77] Y. Lu and J. Han, “Cancer Classification Using Gene Expression Data,” Information Systems, vol. 28, pp. 243–268, 2003.

[78] D. J. C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge, U.K.: Cambridge University Press, 2003.

[79] S. C. Madeira and A. L. Oliveira, “Biclustering Algorithms for Biological Data Analysis: A Survey,” IEEE Trans. on Computational Biology and Bioinformatics, vol. 1, no. 1, pp. 24–45, 2004.

[80] S. N. Mukherjee, P. Sykacek, S. J. Roberts, and S. J. Gurr, “Gene Ranking Using Bootstrapped P-Values,” SIGKDD Explorations, vol. 5, no. 2, pp. 16–22, 2003.

[81] W. Pan, “A Comparative Review of Statistical Methods for Discovering Differentially Expressed Genes in Replicated Microarray Experiments,” Bioinformatics, vol. 18, pp. 546–554, 2002.

[82] G. Piatetsky-Shapiro, T. Khabaza, and S. Ramaswamy, “Capturing Best Practice for Microarray Gene Expression Data Analysis,” in Proc. of the 9th ACM SIGKDD

Int’l Conf. on Knowledge Discovery and Data Mining, Washington, DC, 2003, pp. 407–415.

[83] A. Savasere, E. Omiecinski, and S. Navathe, “An Efficient Algorithm for Mining Association Rules in Large Databases,” in Proc. of the 21st Int’l Conf. on Very Large Data Bases, Zurich, Switzerland, 1995, pp. 432–444.

[84] R. Simon, “Supervised Analysis When the Number of Candidate Features Greatly Exceeds the Number of Cases,” SIGKDD Explorations, vol. 5, no. 2, pp. 31–36, 2003.

[85] E. P. Xing, M. I. Jordan, and R. M. Karp, “Feature Selection for High-Dimensional Genomic Microarray Data,” in Proc. of the 18th Int’l Conf. on Machine Learning, Williamstown, MA, 2001, pp. 601–608.

[86] L. Yu and H. Liu, “Redundancy Based Feature Selection for Microarray Data,” in Proc. of the 10th ACM SIGKDD Int’l Conf. on Knowledge Discovery and Data Mining, Seattle, Washington, 2004, pp. 737–742.

[87] H. Zhang, C. Y. Yu, B. Singer, and M. Xiong, “Recursive Partitioning for Tumor Classification with Gene Expression Microarray Data,” Proc. of the National Academy of Sciences of the United States of America, vol. 98, no. 12, pp. 6730–6735, 2001.

[88] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, Wiley, 2000.

[89] L. Parsons, E. Haque, and H. Liu, “Subspace Clustering for High Dimensional Data: A Review,” ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 90-105, 2004.

[90] P. Berkhin, “Survey of Clustering Data mining Techniques,” Technical report, Accrue Software, San Jose, CA, 2002.

[91] A.K. Jain, M.N. Murty, and P. J. Flynn, “Data clustering: a review,” ACM Computing Surveys, vol. 31, no. 3, pp. 264-323, 1999.

[92] Y. Cheng and G.M. Church, “Biclustering of Expression Data,” Proc. of the 8th Int. Conf. on Intell. Syst. Mol. Biol., La Jolla, California, pp. 93-103, 2000.

[93] S.C. Madeira and A.L. Oliveira, “Biclustering Algorithm for Biological Data Analysis: A Survey,” IEEE Trans. on Computational Biology and Bioinformatics, vol. 1, no. 1, pp. 24-45, 2004.

[94] J. Hipp, U. Gűntzer, and G. Nakhaeizadeh, “Algorithms for Assocation Rule Mining – General Survey and Comparsion,” ACM SIGKDD Explorations Newsletter, vol. 2, no. 1, pp. 58 – 64, 2000.

[95] J. Han, Data Mining: Concepts and Techniques. Morgan Kaufmann, 2001.

[96] P. M. Murph and D. W. Aha, “UCI Repository of Machine Learning Databases,” Dept. of Information and Computer Science, Univ. of California, Irvine, 1987.

[97] J. Ghosh. Handbook of Data Mining, Lawrence Erlbaum Assoc., 2003.

[98] S. Brin, R. Motwani, R. Silverstein, “Beyond Market Basket: Generalizing Association Rules to Correlations,” Proc. ACM SIGMOD Conf. Management of Data (SIGMOD’97), pp. 265-276, 1997.

[99] A. Silberschatz and A. Tuzhilin, “What Makes Patterns Interesting in Knowledge Discovery Systems,” IEEE Trans. on Knowledge and Data Eng., vol. 8, no. 6, pp. 970-974, 1996.

[100] R. Srikant, Q. Vu, and R. Agrawal, “Mining Association Rules with Item Constraints, “ Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining (KDD’97), pp. 67-73, 1997.

[101] R. Bayardo, R. Agrawal, and D. Gunopulos, “Constraint-Based Rule Mining in Large, Dense Databases,” Proc. 15th Int. Conf. Data Engineering (ICDE'99), pp. 188-197, 1999.

[102] B. Liu, W. Hsu, and Y. Ma, “Pruning and Summarizing the Discovered Associations,” Proc. 5th Int. Conf. Knowledge Discovery and Data Mining (KDD’99), pp. 125-134. 1999.

[103] M. Mahta, R. Agrawal, and J. Rissanen, “SLIQ: A Fast Scalable Classifier for Data Mining,” Proc. Fifth Int. Conf. on Extending Database Technology (EDBT’96), 1996.

[104] A. Baraldi and P. Blonda, “A Survey of Fuzzy Clustering Algorithms for Pattern Recognition – Part I,” IEEE Trans. on Syst., Man, Cybern. – Part B, vol. 29, no. 6, pp. 778-785, 1999.

[105] A. Baraldi and P. Blonda, “A Survey of Fuzzy Clustering Algorithms for Pattern Recognition – Part II,” IEEE Trans. on Syst., Man, Cybern. – Part B, vol. 29, no. 6, pp. 786-801, 1999.

[106] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, 1981.

[107] E. Erwin, K. Obermayer, and K. Schulten, “Self-organizing maps: Ordering, Convergence Properities and Energy Functions,” Biol. Cybern., vol. 67, pp. 47-55, 1992.

[108] J. C. Bezdek and N. R. Pal, “Two Soft Relative of Learning Vector Quantization,” Neural Networks, vol. 8, no. 5, pp. 729-743, 1995.

[109] G. Carpenter, S. Grossberg, N. Maukuzon, J. Reynolds, and D. B. Rosen, “Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps,” IEEE Trans. Neural Networks, vol. 3, no. 5, pp. 698-713, 1992.

[110] I. Joliffe, Principle Component Analysis, Springer-Verleg, 1986.

[111] M. W. Berry and M. Browne, Understanding Search Engines: Mathematical Modeling and Text Retrieval, SIAM, 1999.

[112] X. Z. Fern and C. E. Brodley, “Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach,” Proc. 20th Int. Conf. Machine Learning (ICML’03), 2003.

[113] C. K. Chow and C. N. Liu, “Approximating discrete probability distributions with dependence trees,” IEEE Trans. on Information Theory, vol. 14, no. 3, pp. 462-467, 1968.

[114] O. L. Mangasarian and W. H. Wolberg, “Cancer diagnosis via linear programming,” SIAM News, vol. 23, no. 5, pp. 1 & 18, 1990.

[115] J. Zupan, Clustering of Large Data Sets, Research Studies Press, 1982.

[116] I. H. Witten and E. Frank, Data Mining: Practical machine learning tools with Java implementations, Morgan Kaufmann, 2000.