The Hong Kong Polytechnic University

Department of Computing

Unsupervised Pattern Discovery for Sequence and Mixed Attribute Databases

Wu Pak Kit

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF PHILOSOPHY

December 2010

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

(Signed)

Wu Pak Kit

(Name of Student)

Abstract

That the world contains a vast amount of digital information getting ever vaster ever more rapidly, there is a great need to reveal new insights which previously remain hidden from the data of mixed data types such that comprehensive information could be well structured, effectively organized and further applied to analysis, classification, interpretation, understanding and summarization. As most data from databases come from diverse sources, many of them are not necessarily provided with explicit class information. A pattern discovery method which automatically discovers pattern and knowledge from data without relying on prior classificatory knowledge is in great need.

For a large database, how to discover statistically significant patterns and how to discretize its continuous data into interval events are still research and practical problems. Discovering patterns from a large mixed-mode database, where these data types may be a mixture of interval-scaled, symmetric binary, asymmetric binary, category, ordinal or ratio-scaled, is regarded as a classification problem when classes of the samples are given and solved as a discrete-data problem by discretizing the continuous data into intervals maximizing the interdependence between that attribute and the class labels. However, when class information is unavailable, discovering patterns becomes difficult. To tackle the aforementioned problems in an unsupervised manner, which is the problem of

unsupervised pattern discovery, one would search for statistically significant patterns by mining the database. The proposed approach adopts a probabilistic approach to detect statistically significant patterns and transform them into a relational table to represent the original data. Given a mixed-mode dataset, we partition it into a number of attribute clusters, each of which contains some sort of correlated relationship. This process is known as attribute clustering. Once all optimal attribute clusters are found, the most representative attribute so-called mode could be discovered in each attribute cluster. To deal with the discretization problem, a mode-driven discretization algorithm is introduced to treat the mode just like the class label to drive the discretization of other continuous attributes in the attribute group by maximizing the interdependence between the continuous attributes and the mode. Treating intervals as discrete events, association patterns can be discovered. If the attribute clusters obtained are crisp clusters, significant patterns overlapping different clusters cannot be found. A new method of "fuzzifying" the crisp attribute clusters is introduced to detect significant patterns which overlap different fuzzy clusters.

In validating the premises proposed in the thesis, extensive experiments using a number of synthetic data sets, data sets from UCI machine learning archive and two large sets from real world databases were conducted to verify each of the questions conceived.

In particular to demonstrate the usefulness of the proposed approach, the two large sets of real world data are chosen to be analyzed: one is from a number of meteorological surface stations while another one is from a delay coking unit in a petrochemical refinery. The discovery of patterns from the data of weather stations reflects the local and global characteristics of the correlated meteorological parameters. The finding from the data of the delay coking reveals the relationship among the large number of sensors and controllers of the coking plant facilities. These findings provide significant evidences to support the usefulness and effectiveness of the proposed approaches in analyzing the data to extract significant patterns and knowledge for interpretation, understanding and summarization.

Publications Arising from the Thesis

1. Wong A.K.C., Wu B., Wu G.P.K., and Chan K.C.C. "Pattern Discovery for Large Mixed-Mode Databases", In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, Toronto, Canada, 859-868, (October) 2010.
2. Wong A.K.C., Wu G.P.K., Chan K.C.C. and Wu B. "Pattern Discovery on Mixed-Mode Data", Submitted to *IEEE Transactions on Knowledge and Data Engineering*, 2011.
3. Wu G.P.K., Chan K.C.C. and Wong A.K.C. "Unsupervised Fuzzy Pattern Discovery in Gene Expression Data", *BMC Bioinformatics*, 12 (Suppl 5):S5, 2011.
4. Wu G.P.K., Chan K.C.C., Wong A.K.C., and Wu B. "Unsupervised Discovery of Fuzzy Patterns in Gene Expression Data", In *Proceedings of the IEEE International Conference on Bioinformatics & Biomedicine*, Hong Kong, 269-273, (December) 2010.

Acknowledgements

First, 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 Prof. Keith C.C. Chan, Department of Computing, The Hong Kong Polytechnic University, for his valuable help, patient and encouragement during my study.

I would also like to express my deep appreciation to Prof. Andrew K.C. Wong, Department of Systems Design Engineering, University of Waterloo, for his continuous support and help during my study in the M.Phil. program.

Last but not least, I would like to thanks my family for their continuous caring. Without their love, this work is probably not possible.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

In the recent years, with the progress of microelectronics, information technologies together with the ever broadening use of computers in a vast spectrum of business and industry, the volumes of databases have been growing from gigabytes to terabytes as well as to petabytes and the types of data in databases are becoming more and more diverse. Some databases contain either numeric, symbolic or categorical data. Others contain a mixture of the aforementioned types and are referred to as mixed-mode data. Such these data are commonly stored in a relational database with mixed-mode attributes. A sequence database, besides mixed-mode data, is a very significant type of data in many areas such as web log sequence, biological sequence, customer purchasing history, event sequences and so on. A vast amount of these types of data from different areas has been collected. The discovery of new interesting knowledge from them has important applications and creates great value in many sectors.

Today, we are facing problems in handling large sequence and relational databases with mixed-mode attributes and sequence data. Often these sensed or documented data are acquired from different aspects or components of a complex system. Their use is not necessarily confined to classification and often they contain no specific class information, i.e. class labels. Nevertheless, there are great needs to discover patterns from these types of data for the comprehensive analysis, interpretation and understanding of the knowledge inherent in them. This thesis presents new methods for unsupervised pattern discovery that is to discover patterns from large sequence and mixed-mode databases where class information is non-existing or unavailable.

In the past, data mining techniques have been developed mostly for continuous or categorical data. In inductive machine learning, classification information is obtained from a collection of pre-labeled data samples, and thus classification rules or models could be built based on them. Nevertheless, in real world applications, most real databases may be composed of not only continuous data but also of mixed-mode (continuous and discrete). For a learning system to operate under a mixed-mode database, either these continuous attributes may need to be first discretized/quantized into a finite number of intervals, or these discrete attributes may need to be converted to continuous attributes. If the data attributes could be converted appropriately, the limitations of most inductive machine learning systems and algorithms may be solved by feeding in the transformed database instead of the original mixed-mode database. Indeed, in nowadays' data mining, pattern discovery and machine learning development, classification tasks in mixed-mode database require the existence of class labels. With class labels, continuous data attributes could be discretized by the class-dependent discretization, allowing the application of contemporary data mining methodologies. However, in case of class label unavailability, the continuous data could not be handled properly for both supervised learning and unsupervised learning.

In sequential pattern mining, frequently occurring ordered events or subsequences are mined as patterns. Most studies concentrate on categorical patterns while time series analysis focuses on numerical curve analysis. Most relational databases may treat a sequence attribute as text (discrete event) or a sequence database itself contains no other mixed-mode attributes but only a set of tuples, each of which contains a sequence ID and a sequence. The discovered

patterns from sequences could further be used for classification and/or clustering. In classifying sequences, most existing algorithms require the sequences to be aligned first before their tasks. The alignment (substitutions, insertions, and deletions) process consists of placing proper space or removing some items from different sequences in columns to optimize a scoring function. However, this process may create errors as the scoring function may not reflect the true weight in each alignment operation. Without sequence alignment, sequence classification could be operated by making use of some probabilistic measures to determine whether an item is statistically significant to a particular class (i.e. relying on class information). However, unsupervised learning or, more precisely, sequence clustering is difficult to proceed when class information is unavailable.

1.1 Motivations

That the world contains a vast amount of digital information getting ever vaster ever more rapidly [Economist 2010], there is a great need to reveal new insights which previously remain hidden from the data of mixed data types such that comprehensive information could be well structured, effectively organized and further applied to analysis, classification, interpretation, understanding and summarization.

As most data from databases come from diverse sources, many of them are not necessarily provided with explicit class information. A pattern discovery method which automatically discovers pattern and knowledge from sequence based and/or mixed-mode based data without relying on prior classificatory knowledge is in great need. Once such methods are developed, they could be applied to data mining tasks including data clustering [Wong and Wang 1979; Ma, Chan and Chiu 2005; Dias and Cortinhal 2008; Cadez et al. 2003; Alon et al.

1999; Baraldi and Blonda 1999; Jain, Murty and Flynn 1999; Wong, Chiu and Huang 2002; Jiang, Tang and Zhang 2004; Domany 2003; Smet et al. 2002; Zupan 1982; Parsons, Haque and Liu 2004; Berkhin 2002; Madeira and Oliveira 2004; Fern and Brodley 2003; Cheng and Church 2000], pattern discovery [Wong and Wang 1997; Ma, Chan and Chiu 2005; Wong and Wang 2003, Chau and Wong 1999; Wong et al. 2010; Wang and Wong 2010], and other pattern post-processing such as pattern clustering [Wong and Li 2008; Wong and Li 2010] and pattern summarization [Wong and Li 2008; Wong and Li 2010], aiming to discover previously hidden knowledge from data.

In the past decade, pattern discovery methods have been used to obtain classification knowledge to build models for classification and prediction tasks. Later, it attempts to uncover the underlying principles and behaviors of systems or phenomena in the real world from acquired data so as to reason, infer and predict the behaviors. Nowadays, it still poses some challenges as follows.

I. Regarding to the structures of sequence and mixed-mode data space and data subspace, it is becoming more and more complicated than ever before. The values of data could include lengthy sequences or mixed-mode (discrete or continuous) types and the data dimensionality is high. An attribute in a data subspace could be partially a member of other data subspaces. The data collection could be systematic or sometimes arbitrarily.

II. Regarding to the quality of the data, for many reasons, a database might contain noisy data. For a pattern discovery method to be effective, it should be a probabilistic approach rather than a deterministic one.

III. Regarding to *a priori* knowledge, in some situations, it could be difficult to collect adequate correct domain knowledge for effective decision making. Some domain experts are able to support some examples and measurements to formulate a domain database for analysis but also would like to obtain some suggestions or evidences provided by the data analysis outcome for realizing the thoughts and ideas. This raises the focus on unsupervised learning.

IV. Regarding to the application of discovered patterns, some kinds of interestingness measurements such as for pattern support and confidence should be taken into consideration for assisting decision making, interpretation and summarization.

With these motivations in mind, it opens some research challenges and issues to be investigated to the data mining community in the recent years. Naturally, this thesis takes the mentioned issues as the research motivation essentially.

1.2 Objectives

The objectives of this thesis are motivated by the aforementioned practical needs derived from the real world application and are specifically listed as follows.

I. To discover statistically significant sequential patterns from a sequence database: For a large sequence database, different sequences may contain common patterns and/or different patterns. Each pattern could represent a certain real world event. Whether sequences in the database are classified or clustered, statistically

significant sequential patterns must be discovered first in order for further pattern analysis.

II. To transform a sequence database based on discovered patterns into a relational database for further analysis: Once the statistically significant sequential patterns are discovered, the original sequence database will be processed based on the discovered characteristics to construct the transformed database similar to a relational database with continuous attributes.

III. To partition a large mixed-mode database into fuzzy sub-databases (fuzzy attribute clusters) containing attributes with high interdependence with each other in the same data subspace and with fuzzy degrees of membership in the entire data space: For a very large mixed-mode database, different subgroups of the attributes may be governed by different underlying factors or models. These factors or models are defined as the representative attributes (the modes) of subgroups. Each cohesive subgroup in the mixed-mode database could represent a certain aspect of the real world system. With attribute clustering based on the interdependence redundancy measure, the mixed-mode database could be partitioned into some coherent subgroups with strong intra-group interdependence. If data subgroups are crisp clusters, significant patterns overlapping different clusters cannot be found. To close this gap, “fuzzifying” crisp attribute clusters to find patterns in overlapping or fuzzy clusters should be considered.

IV. To discretize continuous attributes in each fuzzy sub-database (fuzzy attribute cluster): Once the mixed-mode database is partitioned into attribute subgroups (clusters), each of which contains attributes with fuzzy degrees of membership to other subgroups, 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 cluster is the representative attribute which is defined as the attribute with the highest total interdependence with others in the same group. Thus class-dependent discretization algorithm could be applied to achieve the task, if the mode is considered to be the implicit class attribute.

V. To apply pattern discovery on a mixed-mode database: Once the continuous data in the mixed-mode database is transformed into discrete interval events, an effective pattern discovery method on categorical data could be applied to each sub-database or to the entire database after all sub-databases re-pool together so as to discover significant event association patterns. Without relying on class information, the patterns discovered in the general form of a subset of categorical, interval data or a combination of both will then become explicit and have a broader spanning coverage on the entire database.

1.3 Organization of this Thesis

This thesis consists of six chapters and is organized as follows:

Chapter 1 introduces the motivation, objective and organization of this work.

Chapter 2 introduces the related knowledge, including the background to sequence clustering, class dependent discretization of continuous data, attribute clustering, pattern discovery and bioinformatics through a literature review.

In Chapter 3, the problem definitions of mining patterns in sequence data and mixed-mode data are presented. The overview of the proposed mining approach is also given. This approach is composed of a collection of techniques, including an unaligned sequence clustering algorithm and an unsupervised pattern discovery algorithm for mixed-mode data.

In Chapter 4, we propose a new approach to cluster sequence data based on the sequential pattern similarity. The proposed approach begins with data conversion based on the sliding window concept to transform the original sequence data. Instead of mining the original sequence data, we mine association patterns from the transformed data. From the patterns discovered in the transformed data, we find clusters involving attributes, which are the hidden patterns, not contained in the original sequence data. To evaluate the effectiveness, the proposed approach is first applied on a synthetic dataset, then on a real-world web log dataset and, finally on a yeast genome sequence dataset. The experimental results show that the proposed approach can produce meaningful clustering results and makes the hidden patterns explicit to build accurate classification/prediction models.

In Chapter 5, we define the problem of unsupervised pattern discovery for mixed-mode data and introduce a methodology for solving it. The proposed method that optimizes some information measures, such as mutual information

and entropy, between attributes of mixed data types groups interdependent attributes into groups/clusters. By applying the proposed algorithm to the mixed-mode database, meaningful grouping of attributes based on interdependence within group helps capture correlated relationship in each group. To reflect the overlapping relationship among attribute clusters, a fuzzy membership function is constructed for each attribute to help capture overlapping relationship in multiple groups. The mode, one with strongest interdependence to the others within group, is used to drive the discretization of continuous data. Combining the information from the fuzzy membership function of each attribute to each mode, patterns in overlapping attribute clusters could be discovered. To evaluate the performance, we applied them on 2 synthetic datasets and several real world datasets. The experimental results of the data mining tasks prove that they are capable of revealing interesting patterns and building very accurate classification/prediction models.

Chapter 6 concludes the thesis, which summarizes its contributions in three aspects including theoretical, methodological and application, and suggests directions for the further research.

Chapter 2. Related Work

In this chapter, the related work in the literature will be surveyed. It will first provide the state of the art of existing data mining techniques in related research areas. It will then present the work related to discretization, attribute clustering, pattern discovery and data mining in bioinformatics in subsequent sections. Different approaches will be discussed in these sections.

2.1 Overview

Knowledge discovery from data (KDD) or data mining, in general, is to extract interesting, nontrivial, implicit, previously unknown and potentially useful information from data [Han and Kamber 2001]. Nowadays, data are everywhere and thus this integrated process is applied in a wide range of sectors such as business, science and engineering. In bioinformatics, this integrated process plays a significant role to discover knowledge in the biological context such as finding motifs in sequences to predict folding patterns, discovering genetic mechanisms underlying a disease, summarizing clustering rules for multiple DNA or protein sequences and more and more. Due to the substantial growth of biological data, data mining or KDD is considered as an important research area in analyzing the data and in solving emerging problems.

This chapter aims at introducing some of the best existing techniques for data mining in related research areas and their applications in bioinformatics in a way that the current research will build on them to make new discoveries. It provides an overview of the work in the literature and how the current work relates to one another.

2.2 Knowledge Discovery and Data Mining in Related Area

According to [Smyth et al. 1996], knowledge discovery in databases or data mining is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. This implicit, previously unknown, and potentially useful information which we call knowledge is hidden in the databases and is usually in the form of relationships among data items. These relationships are possibly in the form of functional, or partial functional dependencies, their discovery analysis and characterization may involve the use of various techniques. The process of applying knowledge discovery in a particular situation consists of the following phases [Han and Kamber 2001]:

1. Understanding the Application domain: This includes the understanding of the relevant prior knowledge and the goals of the application.
2. Extracting the target data set: This includes the selection of a data set or focusing on a subset of variables.
3. Data preprocessing and transformation: This phase improves the quality of the actual data for data mining. It also increases the efficiency of data mining by reducing the computational effort for mining the preprocessed data. Data preprocessing involves data cleaning, data transformation, data integration and data reduction or compression. Data cleaning consists of some basic operations such as normalization, noise removal, handling of missing data, reducing redundancy and etc. Data integration includes integrating multiple and heterogeneous data sets from different data sources. Data reduction finds useful features to represent the data by

means of dimensionality reduction, feature selection, discretization and etc.

4. Data mining: This phase constitutes one or more of the following functions including classification and prediction, association analysis, cluster analysis and etc.
5. Pattern interpretation and evaluation: This phase includes interpreting the discovered patterns and the possible visualization of them. Visualization is important that it increases understandability from the perspective of humans. The mined patterns can be evaluated automatically or semi-automatically to identify the interestingness or usefulness of them.
6. Using discovered knowledge: This phase incorporates the discovered knowledge into the expert system and actions can be taken based on this knowledge.

Knowledge discovery or data mining techniques are used and applied in a wide spectrum of areas. This involves discovering patterns over the data set. Some of the common data mining algorithms will be discussed in the following subsection.

2.2.1 Association, Classification and Clustering

Association analysis mines or generates rules from the data. Association rule mining refers to discovering associations [Cheung et al. 1996; Agrawal and Srikant 1994] among different attributes. It tries to describe the relationship among data items. A population application of association rules mining is the analysis of supermarket transaction data, helping the planning of marketing strategies. Popular algorithms include AIS [Agrawal , Imielinski and Swami

1993], SETM [Houtsma and Swami 1993], and Apriori [Agrawal and Srikant 1994].

Classification analysis classifies a data item into one of several predefined categorical classes. Based on the predefined classes in the training objects, the general approach involves a systematic search for minimal descriptions which can distinguish between members of different classes. In machine learning terminology, this is a type of supervised learning [Tou and Gonzalez 1974], i.e. learning is done with explicit training examples. Popular algorithms include k-nearest neighbor (k-NN) [Dasarathy 1991], decision-tree generators (ID3 [Quinlan 1987], C4.5 [Quinlan 1993], CART [Breiman et al. 1984]), neural networks [Aleksander and Morton 1990; Beale and Jackson 1990] and genetic algorithms [Davis 1991; Goldberg 1989; Holland 1987].

Cluster analysis maps a data item into one of several clusters, where clusters are natural groupings of data items based on distance measure (as known as similarity measure). In general, the resulting clusters should exhibit high within-cluster homogeneity and high between-cluster heterogeneity. Clustering is dependent on the distance measure to be applied. In machine learning terminology, this is a type of unsupervised learning [Tou and Gonzalez 1974], i.e. learning is done without explicit training examples. Commonly, clustering algorithms can be classified into two categories: (1) hierarchical and (2) non-hierarchical. Hierarchical clustering involves the construction of a hierarchy or tree structure. Popular hierarchical clustering algorithms include agglomerative [Milligan 1980], Chameleon [Karypis, Han and Kumar 1999], DIANA [Kaufman and Rousseeuw 1990], AGNES [Kaufman and Rousseeuw 1990] and BIRCH [Zhang, Ramakrishnan and Livny 1996]. Non-hierarchical clustering does not

involve the construction of the tree structure while it first selects a cluster center or seed and then all objects or data points within a pre-specified threshold distance are included in the resulting cluster. Popular non-hierarchical clustering algorithm includes k-means [Forgy 1965; MacQueen 1967], CLARA [Kaufman and Rousseeuw 1990], CLARANS [Ng and Han 1994], CLIQUE [Agrawal et al. 1998] and SOM [Kohonen 1989].

2.2.2 Discretization of Continuous Data

Data discretization techniques can be used to reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals [Han and Kamber 2001]. Discretization is a technique to partition continuous attributes into a finite set of adjacent intervals in order to generate attributes with a small number of distinct values [Tsai, Lee and Yang 2008]. In short, a continuous variable can be discretized into a finite number of discrete intervals. Interval labels can be applied to replace actual value. There are several reasons to perform discretization as a data preprocessing step for data analysis. The obvious reason is it reduces and simplifies the original data, leading to a concise, easy-to-use, and knowledge-level representation of mining results. In data mining algorithms, some have been developed to handle categorical attributes such as AQ [Kaufman and Michalski 1999; Michalski et al. 1986], CLIP [Cios and Kurgan 2001; Cios and Kurgan 2004] and CN2 [Clark and Niblett 1989], while others can deal with continuous attributes but have better performance on categorical attributes [Wu et al. 2006]. Since continuous data can be discretized into a finite set of discrete intervals, discretization can be performed prior to the learning process [Chan, Ching and Wong 1992]. A good

discretization algorithm can produce a concise summarization of continuous attributes but also accounts for learning faster and accurate [Liu et al. 2002].

In some data sets, some of the attributes may be discrete and other attributes may be continuous. Dating back to 1990s, there was no fully integrated approach of inductive learning (IL) which can deal with mixed-mode continuous and discrete data simultaneously [Wong and Chiu 1987]. Ching, Wong and Chan [Ching, Wong and Chan 1995] have proposed a class attribute dependent discretization (CADD) method to deal with mixed-mode data. Two importance decisions must be made for discretization. Firstly, the number of discrete intervals must be selected. Secondly, the width of the intervals must be determined. Their method can automatically determine the most preferred number of intervals to tackle the first decision, and seeks to maximize the mutual dependence between the discrete intervals and class labels to tackle the second decision.

According to [Liu et al. 2002], the discretization algorithms can be classified into five axes: supervised versus unsupervised, static versus dynamic, global versus local, top-down (splitting) versus bottom-up (merging), and direct versus incremental.

1. Supervised methods discretize attributes with the consideration of class information, while unsupervised methods do not.

2. Dynamic methods consider the interdependence among the features attributes and discretize continuous attributes when a classifier is being built. On the contrary, the static methods consider attributes in an isolated way and the discretization is completed prior to the learning task.

3. Global methods, which use total instances to generate the discretization scheme, are usually associated with static methods. On the contrary, local methods are usually associated with dynamic approaches in which only parts of instances are used for discretization.

4. Bottom-up methods start with the complete list of all continuous values of the attribute as cut-points and remove some of them by merging intervals in each step. Top-down methods start with an empty list of cut-points and add new ones in each step.

5. Direct methods, such as Equal Width and Equal Frequency [Chiu, Wong, and Cheung 1991], require users to decide on the number of intervals k and then discretize the continuous attributes into k intervals simultaneously. On the other hand, incremental methods begin with a simple discretization scheme and pass through a refinement process although some of them may require a stopping criterion to terminate the discretization.

More detailed discussion about the five axes mentioned above can be found in [Liu et al. 2002]. In this section, the discussion of discretization algorithms will follow the axis of top-down versus bottom-up.

Class-Attribute Contingency Coefficient (CACC) by [Tsai, Lee and Yang 2008] is one of the latest top-down discretization algorithms. The main contribution of it is that it can generate a good discretization scheme and its discretization scheme can lead to the improvement of classifier accuracy like that of C5.0. The quality of a discretization scheme can be measured by Class-Attribute Interdependence Redundancy (CAIR) proposed by [Ching, Wong and Chan 1995]. The general goal of a discretization to achieve is: 1) a high quality discretization scheme to help users understand the data easily, 2) the scheme

should lead to the improvement of accuracy and the efficiency of a learning algorithm which is the training time and the number of rules generated to reach the classification accuracy, and 3) the discretization process should be as fast as possible. Class-attribute Interdependence Maximization (CAIM) by [Kurgan and Cios 2004] is another top-down discretization algorithm with good performance in comparison with seven state-of-the-art top-down discretization algorithms. On average, experiments show that CAIM obtains high CAIR value, and using it as a preprocessor for classification algorithm, it produces the least number of rules and reach the highest classification accuracy [Kurgan and Cios 2004].

Top-down (splitting) and bottom-up (merging) discretization algorithms consists of unsupervised and supervised. Two typical unsupervised top-down algorithms are Equal Width and Equal Frequency by [Chiu, Wong, and Cheung 1991]. Other the state-of-the-art supervised top-down algorithms are Paterson-Niblett [Paterson and Niblett 1987], maximum entropy [Wong and Chiu 1987], Information Entropy Maximization [Fayyad and Irani 1993], Class-Attribute Dependent Discretizer (CADD) [Ching, Wong and Chan 1995], Class-Attribute Interdependence Maximization (CAIM) [Kurgan and Cios 2004], Fast Class-Attribute Interdependence Maximization (FCAIM) [Kurgan and Cios 2003] and Class-Attribute Contingency Coefficient (CACC) [Tsai, Lee and Yang 2008]. FCAIM has been proposed as a faster version of CAIM extension. The discretization criterion, the stopping criterion and the time complexity between them are the same while the only difference is the initialization of the boundary point. FCAIM was faster than CAIM with similar C5.0 classification accuracy where CAIM obtained a slightly better CAIR value [Kurgan and Cios 2003]. Experiments in [Tsai, Lee and Yang 2008; Kurgan and Cios 2004] showed that

CAIM and CACC are superior to other top-down discretization algorithms as their discretization schemes can generally maintain higher interdependence between target class (also called class label) and discretized attributes, generate lesser number of rules to attain higher classification accuracy. That the abovementioned supervised discretization algorithms aim at seeking local optimal solution, optimal class dependent discretization (OCDD) [Liu, Wong and Wang 2004] searches for global optimum discretization scheme which is proven to be an effective approach experimentally. It is based on the concept of dynamic programming. The current work is adapted from OCDD which searches for the best partition from all possible settings for each iteration.

Four famous bottom-up algorithms are ChiMerge [Kerber 1992], Chi2 [Liu and Setiono 1997], Modified Chi2 [Tay and Shen 2002] and Extended Chi2 [Su and Hsu 2005]. Since bottom-up (merging) algorithms start with all continuous values and recursively remove points by merging intervals, the computational complexity is generally higher than top-down (splitting) algorithms. To merge adjacent intervals, the significant test is performed to test whether or not two adjacent intervals should be merged. Another requirement is that some parameters need to be specified by users such as the significant level, maximal and minimal intervals and etc. Using these bottom-up approaches as preprocessors for C5.0 classification, experiments by [Su and Hsu 2005] showed that Extended Chi2 outperformed the other bottom-up discretization algorithms as its discretization scheme can reach the highest accuracy on average.

In this thesis, we adopt the supervised versus unsupervised discretization category. Up until now to the best of our knowledge, supervised discretization algorithms are generally with better performance than unsupervised

discretization algorithms due to the reason that the supervised one benefits by a prior knowledge.

2.2.3 Attribute Clustering

Attribute clustering also known as feature subspace clustering is to partition a database into a number of sub-databases (attribute clusters) where attributes in the same sub-database have high interdependence or are more relevant while attributes in different sub-databases have low interdependence or are more irrelevant. Attribute is the raw input attribute and the terminology used in relational algebra and relational database while features are attributes constructed for the input attributes. In data mining context, both attributes and features refer to as columns in a dataset. We use without distinction the terms attribute and feature. When machine learning was first introduced, researchers targeted a relatively small set of attributes. As the size of databases and the diversity of attributes increased, data clustering began to break down though the classification problems were not seriously affected yet their effectiveness was diminishing. In supervised learning, the problems were partly solved through feature selection in which it selects a subset of features to represent the whole data. Later, as data mining and pattern discovery came into play, the dimensionality problems were a little relaxed yet the ultimate problem of it still prevailed. Even up to today, most of conventional clustering algorithms will often face the challenges related to the nature of large scale mixed-mode database with a large number of attributes.

In unsupervised learning, attribute clustering was proposed to provide a partial solution as a remedy to the problems but in general, class-dependent discretization had to be used to convert the continuous data into interval data [Au

et al. 2005]. [Mitra et al. 2002] proposes an unsupervised feature selection algorithm based on measuring similarity between features whereby redundancy is removed. Although this method is unsupervised and fast in computation due to its similarity measure based on pair-wise feature, it can only deal with numeric features. To cluster or select attributes, the t-value method is widely used [Agrawal et al. 1992]. It is important to note that the t-value can only be used when the samples are pre-classified. If no class information is provided, it cannot be used for attribute selection. So the Attribute Clustering Algorithm, ACA [Au et al. 2005] was proposed to cluster attributes. In ACA, however, continuous data have to be converted into interval data before attribute clustering could be applied. To close this gap also, this thesis extends ACA so that it is able to deal with mixed-mode data by introducing attribute interdependence redundancy measures between attributes of various attribute types and a multiple interdependence measure [Alon et al. 1999; Wong, Chiu and Huang 2002] for selecting attributes with the highest correlation with the rest of attributes within an attribute cluster. After turning all the continuous data into categorical data, pattern discovery [Wong and Wang 2003], pattern clustering [Wong and Li 2008; Wong and Li 2010] and pattern summarization [Wong and Li 2008; Wong and Li 2010] can then be applied to each of the attribute groups or to the dataset obtained after their re-merging.

2.2.4 Pattern Discovery

Today, pattern discovery for intelligent decision support, knowledge-based reasoning, and data analysis applies more and more to large scale complicated systems and problem domains [Chiu, Wong and Cheung 1991]. In most of the existing systems, data preprocessing, such as data cleansing, filtering,

attribute reduction, are included so as to remove noises, to bring out more relevant information from the data and to reduce the search space. However, they often depend on prior knowledge, such as parameters and preconceived classificatory framework. Thus, they could be very biased and usually involve long iterative search and examination process. To respond to these needs, a data-driven pattern discovery approach has been advanced [Wang and Wong 1979]. It is able to discover, in an unbiased manner, statistically significant events automatically, and to generate decision rules for categorization and prediction. In general, pattern discovery [Wong and Wang 2003] extracts previously unknown regularities in the data and is a useful tool for categorical data analysis.

In most real world problems, pattern discovery typically produces an overwhelming number of patterns, resulting in very difficult and time-consuming effort for problem comprehension and interpretation. To combine fragments of information from individual patterns to produce more generalized forms of information and to use them to further explore or analyze the data, pattern clustering [Wong and Li 2008; Wong and Li 2010] is developed to simultaneously cluster the discovered patterns and their associated data. Pattern summarization [Wong and Li 2008; Wong and Li 2010] can be applied as pattern post-processing method to select from the discovered patterns a most representative subset which could be considered as the summary of the pattern cluster. Once these steps are completed we know how patterns relate locally and how pattern groups are realized in data subspaces and related in the entire data space.

2.3 Data Mining in Bioinformatics

This section will describe some basic concepts of molecular biology for bioinformatics. Then some related work of data mining to the biological domain will be described.

2.3.1 Basic Concepts of Molecular Biology for Bioinformatics

Bioinformatics is the computational branch of molecular biology. Molecular biology concerns itself with understanding the interactions between the various systems of a cell, including the interactions between DNA, RNA and protein biosynthesis as well as learning how these interactions are regulated.

The central dogma of molecular biology was first enunciated by Francis Crick in 1958 and re-stated in a Nature paper published in 1970. The dogma is a framework for understanding the transfer of sequence information between sequential information-carrying biopolymers in living organisms. There are 3 major classes of such biopolymers: DNA and RNA (both nucleic acids), and protein. The process is shown in Figure 2.1.

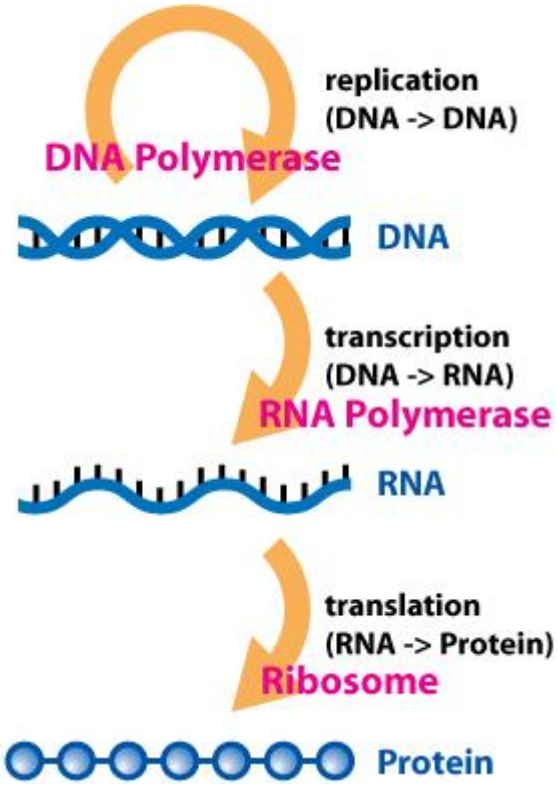


Figure 2.1. Central dogma of molecular biochemistry with enzymes

Figure 2.1 briefly describes the sequential processes from DNA to RNA via a process called transcription and another process from RNA to Protein called translation.

2.3.1.1 DNA

Deoxyribonucleic acid (DNA) is a nucleic acid that contains the genetic instructions used in the development and functioning of all known living organisms and some viruses. The main role of DNA molecules is the long-term storage of information. DNA is often compared to a set of blueprints or a recipe, or a code, since it contains the instructions needed to construct other components of cells, such as proteins and RNA molecules. DNA is a double helix molecule, consisting of two strands of phosphate and deoxyribose sugar molecule. The building blocks of DNA are nitrogenous bases which are adenine (A), guanine (G), cytosine (C) and thymine (T). Each deoxyribose sugar molecule is attached

to one of the above bases. The whole stretch of DNA is called the genome of the organism. The DNA segments that carry this genetic information are called genes, but other DNA sequences have structural purposes, or are involved in regulating the use of this genetic information.

2.3.1.2 RNA

Ribonucleic acid (RNA) is a type of molecules that consists of a long chain of nucleotide units. RNA is very similar to DNA, but differs in a few important structural details: in the cell, RNA is usually single-stranded, while DNA is usually double-stranded; RNA nucleotides contain ribose while DNA contains deoxyribose; and RNA has the base uracil (U) rather than thymine (T) that is present in DNA. RNA is transcribed from DNA by enzymes called RNA polymerases as shown in Figure 2.1. A type of RNA called messenger RNA (mRNA) carries information from DNA to structures called ribosomes for protein synthesis. There are many RNAs with other roles – in particular regulating which genes are expressed, but also as the genomes of most viruses.

2.3.1.3 Protein

Proteins are essential parts of organisms and participate in every process within cells. Many proteins are enzymes that catalyze biochemical reactions and are vital to metabolism. Protein synthesis known as translation is a multi-step process, beginning with amino acid synthesis and transcription which are then used for translation. This uses an mRNA sequence as a template to guide the synthesis of a chain of amino acids that form a protein.

2.3.1.4 Gene Expression

A gene is a sequence of DNA that codes for an RNA. In non-protein coding genes such as rRNA genes or tRNA genes, the product is a functional RNA. In protein coding genes, the RNA in turn codes for a protein. The process by which a gene gives rise to a protein is called gene expression [Lewin 2004]. Figure 2.1 shows this conversion. A gene, however, is not directly translated into protein, but is expressed via the production of an mRNA, a nucleic acid intermediate actually used to synthesize a protein. As the quantity of genes expressed in cells is relevant to the amount of mRNA produced in the process of transcription, measuring the amount of mRNA indirectly measures the expression levels of genes.

2.3.1.5 DNA Microarray Technology

DNA microarrays technology is introduced by [Schena et al. 1995] with the aim to study various molecular mechanisms caused by different genes in cells. It consists of an array of series of thousands of microscopic spots of DNA oligonucleotides, called features. This can be a short section of a gene or other DNA element that are used as probes, attached to a gene chip, to hybridize a cDNA or cRNA sample (called target) under high-stringency conditions. Probe-target hybridization is usually detected and quantified by detection of fluorophore-, silver-, or chemiluminescence-labeled targets to determine relative abundance of nucleic acid sequences in the target. This technology can be used to measure changes in expression level, to detect single nucleotide polymorphisms (SNPs), in genotyping or in resequencing mutant genomes.

To measure gene expression levels, there are two popular kinds of microarray technologies. They are spotted microarray and oligonucleotide

microarray. In spotted microarrays, the probes are oligonucleotides, cDNA or small fragments of polymerase chain reaction (PCR) products that correspond to mRNAs. The probes are synthesized prior to deposition on the array surface and are then spotted onto glass. In oligonucleotide microarrays, the probes are short sequences designed to match parts of the sequence of known or predicted open reading frames. One technique used to produce oligonucleotide arrays include photolithographic synthesis on a silica substrate where light and light-sensitive masking agents are used to build a sequence one nucleotide at a time across the entire array [Pease et al. 1994]. Each applicable probe is selectively "unmasked" prior to bathing the array in a solution of a single nucleotide, then a masking reaction takes place and the next set of probes are unmasked in preparation for a different nucleotide exposure. After many repetitions, the sequences of every probe become fully constructed.

The above mentioned two microarray technologies use different criteria to measure gene expression levels, but they share similar experimental procedures. Figure 2.2 shows the procedures for DNA microarray experiment. The experiment begins by identifying a set of genes to be probed and extracting the corresponding mRNA from this pair of cells. It follows a number of steps including purification, reverse transcription (RT) and coupling to prepare for the labelled samples. The labelled samples are mixed with a hybridization solution and added to a gene chip. The gene chip is scanned in a special machine and a quantitative image is generated with match and mismatch probes. The image is normalized and the hybridization intensity value is converted to represent the gene expression levels in numerical values.

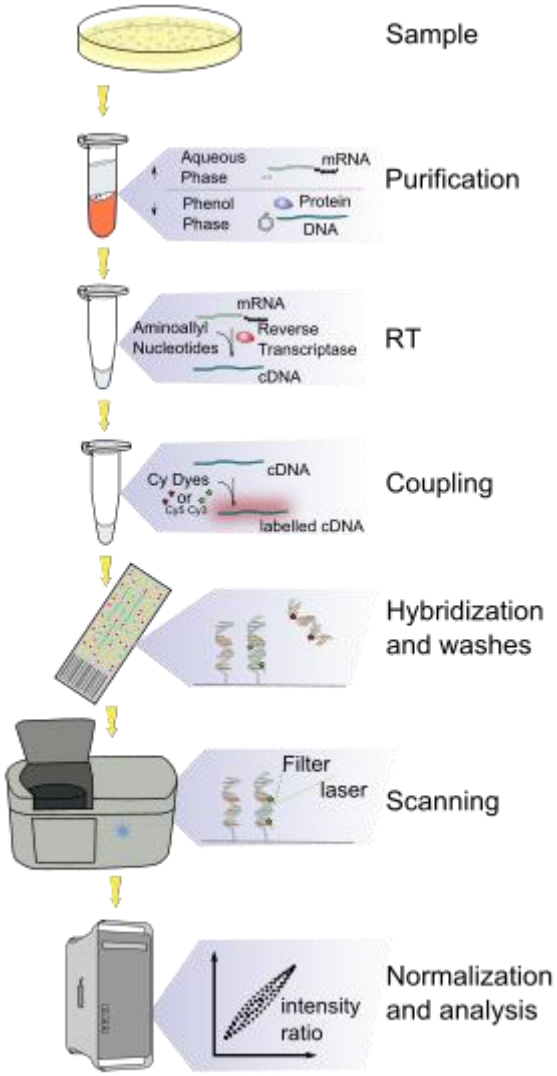


Figure 2.2. Procedures for DNA microarray experiment

The experiment can be repeated under the same experimental condition with different samples from the same type of tissues, resulting in a data set for gene expression data under this experimental condition. Figure 2.3 shows the sample gene expression data set of yeast genome data in a spreadsheet format.

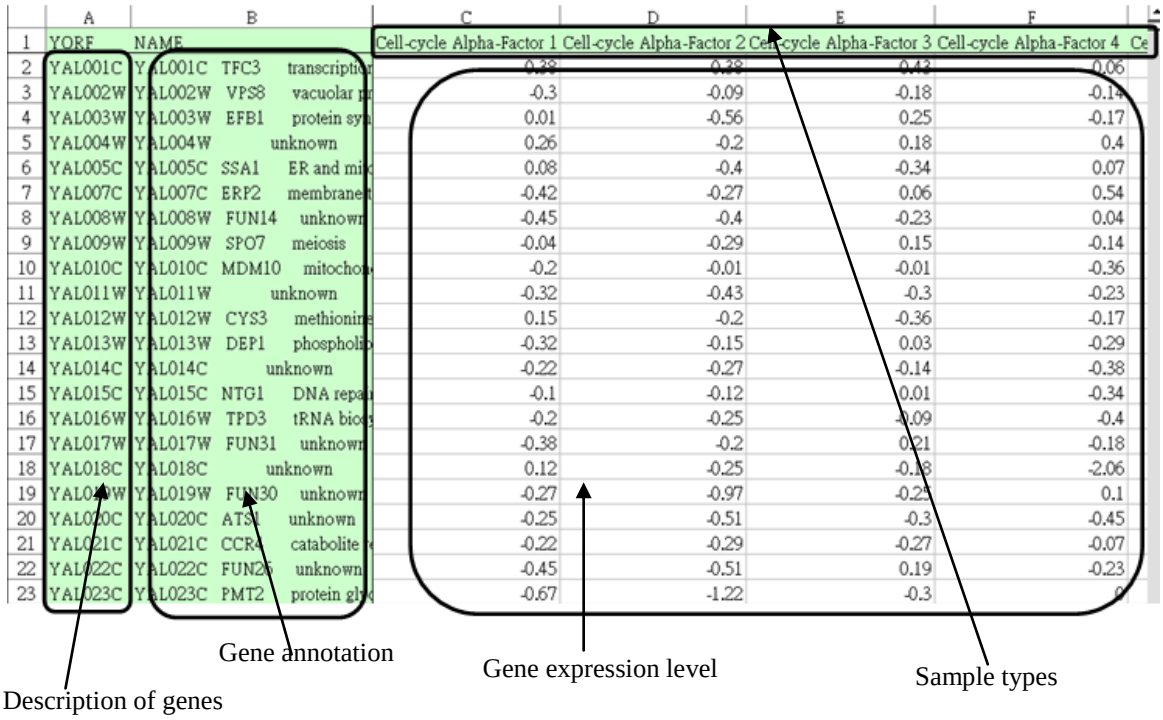


Figure 2.3. A sample gene expression data

Figure 2.3 is a sample gene expression data for yeast genome in a spreadsheet format. It is represented as a gene expression matrix in which each of *n* rows consists of *m*-element expression vector for a single gene.

2.3.2 Data Mining Process for Bioinformatics

Data analysis and mining is a main issue in microarray transcription profiling with the focus on algorithms and database development [Cordero, Botta and Calogero 2008]. The analysis of biological data using clustering and classification techniques has been shown to be useful for predicting biological functions and discovering interesting knowledge [Han and Kamber 2001; Zhang 2006]. The consequent extraction of biological knowledge is also important.

From the studies of [Cordero, Botta and Calogero 2008], the microarray data analysis, which is a process, can be generalized into four steps with different computational tools applied in each step. The first step is quality control. It deals

with the microarray platform chosen to use and the evaluation of the homogeneity of experimental groups. Following by that, it is the data pre-processing. It is the process to transform the raw fluorescence signal detected by microarray technologies into normalized data. Thirdly, it is differential expression detection. This step requires appropriate statistical methods and algorithms to detect and highlight the useful subset of data to further investigate. Much work has been done on this scope in computing community and is the focus of this study. The final step is biological knowledge extraction. Even though the development of techniques for accurate identification of differentially expressed genes, the main difficulty task is in interpretation of them. Recent efforts have shifted from the discovery of gene functions to that of biological pathways, providing a more comprehensive view of the gene expression.

2.3.3 Sequence Clustering

Sequence analysis models sequential patterns, like DNA and protein sequences. Its goal is to model the states of the process generating the sequence. In bioinformatics, sequence clustering algorithms attempts to group related sequences. It is believed that DNA sequence clusters are often synonymous but not identical to biological functional units. In gene expression regulations, understanding the genome sequence is the fundamental step to understand the complex mechanism of gene expression. In clustering biological sequences, a distance measure (as known as similarity score) is often based on sequence alignment. Once the distance measure is chosen, most clustering algorithms such as partitional based approaches and hierarchical based approaches can be adopted. Sequence alignment is a process to properly place conserved residues from different sequences in common derived from common ancestral residues. The

procedure is a hypothetical model of mutations including substitutions, insertions and deletions that had occurred during evolution. However, since the probability of occurrence of different types of mutation is still questionable [Eidhammer 2004], even the best alignment cannot be established unambiguously [Pevsner 2003]. In this regard, the sequence alignment is an error-prone process [Ma and Chan 2008]. To deal with this issue, [Ma and Chan 2008] proposes a classification algorithm UPSEC which is able to compute the similarity score between protein sequences without the need for the alignment of sequences. The hidden patterns of protein classes can be discovered using a probabilistic measure which can determine whether or not a residue is useful for the characterization of a particular class. However, this algorithm is a supervised one which requires the sequences to be pre-labeled classes first before the pattern discovery can begin. In our study, we aim at defining a new technique to cluster biological sequences without a prior knowledge.

2.3.4 Gene Expression Data Analysis

Most living organism contains trillions of cells, each of which carries the same genome. In any given cell, only parts of the genes coded by genome are active. These active genes are “expressed” for the function of a gene. Gene expression is normally referred to as the transcription of mRNA (see Figure 2.1). Gene expression level changes over time in accordance with environmental stimuli. Understanding the expression levels of mRNA under different conditions and over time, it is possible to infer extensive information about gene functions, gene regulations, and gene interactions. DNA microarrays are currently the most popular technology used to measure gene expression level (see 2.3.1.5 for details). The microarray dataset is often represented as gene expression table

$T = \{e_{ij} | i = 1, \ldots, G, j = 1, \ldots, M\}$, where $e_{ij} \in \mathcal{R}$ is the measured expression level of gene g_i in sample s_j, containing rows of genes and columns of samples (see Figure 2.3). The expression values of a gene across different samples are called the gene expression profiles while these of a sample across different genes are called the sample expression profiles.

Two major tasks of gene expression data analysis are classification and clustering. Classification of this type of data is to assign memberships to samples/genes based on expression patterns/profiles while clustering is to find new biological classes/labels and refine existing ones [Piatetsky, Khabaza and Ramaswamy 2003]. In clustering of gene expression data, a typical problem is to handle the high dimensionality space since gene expression data sets normally consists of a large number of genes but a small number of samples.

Due to the characteristic of the high dimensionality of gene expression data, one would cluster both genes and samples [Jiang, Tang and Zhang 2004]. Using conventional clustering methods, the genes/samples are considered as the tuples and the samples/genes as the attribute. Therefore, it is able to identify genes/samples with similar expression patterns (i.e. co-expressed genes/new biological classes). In this way, various existing clustering algorithms could be applied to gene expression data. Popular clustering algorithms are *k*-means [MacQueen 1967], Kohonen's self-organizing maps [Tamayo et al. 1999], and hierarchical based clustering algorithms [Alon et al. 1999; Eisen et al. 1998]. Biclustering algorithms [Madeira and Oliveira 2004; Cheng and Church 2000] have been proposed to cluster both genes and samples simultaneously, besides clustering genes and samples separately. These biclustering algorithms group a

subset of genes and a subset of samples into a bicluster in which the genes and samples exhibit similar behavior.

In analyzing the gene expression dataset, one would measure the similarity between two expression profiles. The similarity measure/distance measure is the critical component in this analysis. Two common similarity measures for gene expression data are Pearson correlation coefficient and Euclidean distance. The Pearson correlation coefficient between two gene expression profiles g_x and g_y can be calculated as:

$$p(g_x, g_y) = \frac{1}{M}\left\{\sum_{j=1}^{M}\left(\frac{e_{xj} - \overline{E_x}}{\theta_x}\right)\left(\frac{e_{yj} - \overline{E_y}}{\theta_y}\right)\right\},$$

where

$$\theta_x = \sqrt{\sum_{j=1}^{M}\frac{(e_{xj}-\overline{E_x})^2}{M}},\ \theta_y = \sqrt{\sum_{j=1}^{M}\frac{(e_{yj}-\overline{E_y})^2}{M}},\ \overline{E_x} = \frac{\sum_{j=1}^{M} e_{xj}}{M},\ \overline{E_y} = \frac{\sum_{j=1}^{M} e_{yj}}{M},$$

M is the total number of experimental conditions (columns) and e_{xj} is the gene expression value of gene x under experimental condition j. The Euclidean distance between two gene expression profiles g_x and g_y can be calculated as:

$$d(g_x, g_y) = \sqrt{\sum_{j=1}^{M}(e_{xj} - e_{yj})^2}.$$

However, an empirical study [Heyer, Kruglyak and Yooseph 1999] has shown that Pearson correlation coefficient is not robust to outliers and it may assign high similarity score to a pair of dissimilar genes. Euclidean distance is the most utilized measure for comparing expression profiles for both genes and samples. It assumes that the gene profiles are uncorrelated so it leads to spherical shaped clusters. Therefore, it is used as a distance measure for many clustering algorithms (hierarchical and partitional) of gene expression data. In biclustering,

a popular measure of the coherence of genes and samples is the mean squared residue [Cheng and Church 2000]. Let $I \subseteq \{1, \ldots, G\}$ and $J \subseteq \{1, \ldots, M\}$. The mean squared residue of a bicluster $T_{IJ} = \{e_{ij} | i \in I, j \in J\}$ is defined as:

$$d_R(T_{IJ}) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (e_{ij} - e_{iJ} - e_{Ij} + e_{IJ})^2,$$

where e_{iJ} is the mean of e_{ij}, $j \in J$, e_{Ij} is the mean of e_{ij}, $i \in I$, and e_{IJ} is the mean of e_{ij}, $i \in I, j \in J$. If the mean squared residue is less than or equal to a user-defined threshold, a bicluster is formed.

Another important analysis of gene expression data is gene selection. Gene selection is used to further narrow down the number of attributes prior to data mining. In the literature [Piatetsky, Khabaza and Ramaswamy 2003], *t*-value is widely used for this purpose. Suppose we are given a gene expression data set with 2 classes of samples, the *t*-value for a gene g_x is given by:

$$t(g_x) = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}},$$

where μ_r and σ_r are the mean and the standard deviation of the expression levels of gene g_x for class *r*, respectively, and n_r is the number of samples in class *r* for r = 1, 2. The genes ranked high can be selected for data mining. When the number of classes is greater than 2, the *t*-value is typically computed for one class versus all the other classes. This method is weak to deal with redundant genes [Ding and Peng 2003]. To handle the redundancy, [Ding and Peng 2003; Yu and Liu 2004] have proposed methods that can handle both gene-class relevance and gene-gene redundancy, using some metrics to measure the gene-class relevance and gene-gene redundancy such as mutual information and information gain. [Au et al. 2005] has proposed an attribute clustering algorithm for optimally selecting genes using an interdependence redundancy measure.

However, the aforementioned methods require domain knowledge or class information to select subsets of genes. In our study, we aim at selecting attributes without requiring the use of class information but it will not scarify a lot of the quality of results.

Overall, there are some challenges in dealing with gene expression data: 1) The dataset is normally in high dimensional space. 2) The presence of both biological and technical noise inherent in the data. 3) The clustering structure is usually unknown. 4) Most distance measures only locally compare pair-wise distances of two genes/samples. 5) The clustering results could be difficult to interpret. Although hierarchical based clustering algorithms return tree structure clusters, users are required to decide the number of clusters based on their domain knowledge to distinguish various clusters. For *k*-means based algorithms, users need to specify the number of clusters initially. Also, the clustering results of genes/samples need to be further analyzed for uncovering the underlying patterns of items in the clusters.

2.3.4.1 Application of Fuzzy Logic in Gene Expression Data Analysis

To mine and analyze unlabeled data with uncertain grouping is fuzzy clustering, that search for natural structure of data with uncertain assignment of clusters. A partition of *n* data samples into *C* clusters is defined by a partition matrix $U = \{u_{ik}\}$, where $0 < u_{ik} < 1$ is the degree of data sample x_k belonging to a cluster *i* subject to a constraint that the total degree of a sample belonging to all clusters being one which is

$$\sum_{i=1}^{C} u_{ik} = 1 \text{ for all } k.$$

We also call u_{ik} the degree of membership of sample x_k in cluster *i*. In crisp clustering case, each x_k is assigned to one and only one cluster *i*. That is, for each $k = 1, \ldots, n$, $u_{ik} = 1$ for some *i* between 1 and *C* and, $u_{jk} = 0$ for all other cluster indices *j*.

Fuzzy C-means clustering algorithm has been employed for gene expression data analysis extensively [Gasch and Eisen 2002; Wang et al. 2003; Belacel et al. 2004; Asyali and Alci 2005]. The use of fuzzy clustering of gene expression data for extracting biological insights is to identify overlapping clusters of genes to observe the response of cells to environmental changes. It has an advantage over crisp clustering due to the fact that gene expression data contains great amount of imprecision and uncertainty. To successfully employ this technique to gene expression data clustering, some issues have to be tackled including algorithm initialization, sensitiveness to noise and outliers, convergence often to a local minimum, and choosing the fuzzy parameter.

The Fuzzy C-means [Bezdek et al. 1999], which is a fuzzy version of *k*-means, is a scheme used to partition a set of data into a predefined number of clusters considering the uncertainty of cluster assignment. It allows for sharing of objects between clusters. Each cluster is represented by a cluster center. Assuming that the data set contains *C* cluster centers and *n* samples. The membership values of a sample x_k to a certain cluster i is

$$u_{ik} = \frac{\left(\frac{1}{d(x_k, v_i)}\right)^{\frac{2}{m-1}}}{\sum_{j=1}^{C}\left(\frac{1}{d(x_k, v_i)}\right)^{\frac{2}{m-1}}},$$

where $i \epsilon C, k \epsilon n$, v_i is the center of cluster *i*, $d(x_k, v_i)$ is the distance between the sample x_k and v_i, m is the fuzzy parameter called the fuzzifier in which a large

value of it favors more fuzzy partitions. Different choices of distance function provide alternate definitions of closeness of objects for clustering approaches.

Chapter 3. The Proposed Approach

In this chapter, the problems of unsupervised mining of patterns in sequence data and mixed-mode data are defined. A new mining approach is proposed to solving them. The proposed approach consists of a collection of techniques for 1) discovering statistically significant patterns from sequence data automatically; 2) using the discovered patterns to construct a transformed relational database to represent the original sequence database for further analysis; 3) grouping, selecting and fuzzifying a subset of attributes of a mixed-mode database for data mining; and 4) enabling pattern discovery involving attributes that are not originally in event level. This chapter will also describe how these techniques integrating into the proposed approach.

3.1 A Formal Problem Description

Suppose that there is a set of unlabelled N sequences of varying length in a sequence database S. Each sequence is represented as $S_i = \{a_{i1}, \ldots, a_{il}, \ldots, a_{iL_i}\}$, where $a_{il} \in \vartheta$ is an item, element, or ordered event, $i = 1, \ldots, N$, $l = 1, \ldots, L_i$, and ϑ is the set of alphabets that an item in a sequence can take on. Thus, L_i is the length of sequence S_i. A sequence S_i with length L_i is called a L_i -sequence. A subsequence $\beta = \{b_1, b_2, \ldots, b_n\}$ of a sequence $S_i = \{a_{i1}, \ldots, a_{il}, \ldots, a_{iL_i}\}$ is denoted as $\beta \sqsubseteq S_i$, if there exist integers $1 \leq j_1 < j_2 < \ldots < j_n \leq L_i$ such that $b_1 \subseteq a_{ij_1}$, $b_2 \subseteq a_{ij_2}$, …, $b_n \subseteq a_{ij_n}$.For example, if β = {“keyboard”, “mouse”} and S_i = {“keyboard”, “mouse”, “speaker”}, then β is a subsequence of S_i and S_i is a supersequence of β.

3.1.1 Unsupervised Mining of Patterns in Sequence Data

From S_1, …, S_i, …, S_N, we aim at mining a set of sequential patterns that are statistically significant to reveal the underlying regularities hidden in the sequence database. For instance, in a customer purchase database, an example of a sequential pattern is β = {"keyboard", "mouse"} which could be interpreted as "*Customers who buy a keyboard are likely to buy a mouse in a transaction.*" A sequence or subsequence is said to be a sequential pattern (or simply a pattern) if it is associated with one or more interestingness measures. A popular kind of interestingness measures is the minimum support threshold [Agrawal , Imielinski and Swami 1993]. The support of a sequence β in a sequence database S is the number of sequences in the database containing β, that is, $support_S(\beta) = |\{S_i | S_i \in S \wedge (\beta \sqsubseteq S_i)\}|$. Given a positive integer *min_sup* as the minimum support threshold, a sequence β is frequent in a sequence database S if $support_S(\beta) > min\_sup$. A frequent sequence is called a sequential pattern (or simply a pattern) based on the minimum support threshold. A pattern with length l is called an *l*-pattern. There are also other types of interestingness measures such as Dempster-Shafer measure [Dempster 1967], conviction [Brin et al. 1997], J-measure [Smyth and Goodman 1992], chi-squared measure [Brin, Motwani and Silverstein 1997], the adjusted residual and weight of evidence [Chan and Wong 1990, Chan and Wong 1991], etc. On the other hand, the discovered patterns should make the characteristics of sequences explicit. Rather than listing the patterns, a transformed relational database with attributes of patterns and tuples of sequence characteristics should be constructed for further data analysis, which is the first goal. Another goal of the problem, unaligned sequence clustering, is to partition these sequences, S_1, …, S_i, …, S_N, into clusters, $C = \{c_1, \ldots, c_k\}$, according to their sequence similarities.

3.1.2 Unsupervised Mining of Patterns in Mixed-Mode Data

We are also concerned with mining a set of mixed-mode data to reveal patterns in vertical and horizontal data space. Consider a data set *D* containing a set of *N*-tuples of mixed-mode data. Every tuple is described by *N* attributes, some assuming discrete values and others continuous values. Let $\boldsymbol{X} = \{X_1, \ldots, X_N\}$ represent this attribute set. For convenience, let us permute the attributes (without affecting the analysis) that the first *M* attributes $\{ Xi | 1 \leq i \leq M\}$ are discrete valued and the remaining $\{Xi' | M+1 \leq i' \leq N\}$ are continuous valued. Then, each X_i, $1 \leq i \leq M$ can be seen as a discrete random variable taking on values from its alphabet $\alpha_i = \{\alpha_i^1, \ldots, \alpha_i^{m_i}\}$, where m_i is the cardinality of the alphabet of the i^{th} attribute. Each X_i, $M+1 \leq i \leq N$ can then be seen as a continuous random variable. Thus, a realization of $\boldsymbol{X}$ can be denoted by $x_k = \{x_{1k}, \ldots x_{ik} \ldots x_{Mk}, x_{(M+1)k}, \ldots x_{i'k} \ldots x_{Nk}\}$ where $\{ x_{ik} | 1 \leq i \leq M\}$ can assume any value in α_i and $\{ x_{i'k} | M+1 \leq i' \leq N\}$ can assume any value in $\{ M_{i'k} \leq \Re \leq N_{i'k}\}$ where $\Re$ is the real number. Thus, each tuple in the data set is a realization of $\boldsymbol{X}$. Based on the problems introduced by Wong et. al. [Wong and Liu 1975; Wong and Wang 1979; Wong and Chiu 1987], database partitioning, by which a database is clustered into interdependent attribute groups first, and data clustering is then applied to each attribute group which contains interdependent attributes only, is extended to study overlapping relationship among attribute groups. Thus, patterns which may overlap crisp attribute clusters could be found within overlapping or fuzzy clusters. This challenging problem, unsupervised pattern discovery for mixed-mode data, will be taken into serious consideration.

Another important problem encountered during pattern discovery or data mining is the discretization of continuous data in the mixed-mode data space. This problem will be addressed in detail in the latter section of the thesis.

3.2 The Solution

Given a dataset of sequence based or mixed-mode based, we propose to use a new data mining approach for the discovery of patterns. To solve the problems: I) unaligned sequence clustering, and II) unsupervised pattern discovery for mixed-mode data, the proposed approach comprises of a collection of techniques for: 1) sequence conversion, 2) interesting association pattern discovery on sequences, 3) clustering and re-clustering, 4) mixed-mode attribute clustering, 5) attribute cluster fuzzification, 6) discretization of continuous data, and 7) pattern discovery. Figure 3.1 shows the proposed data mining approach and how these techniques integrate together.

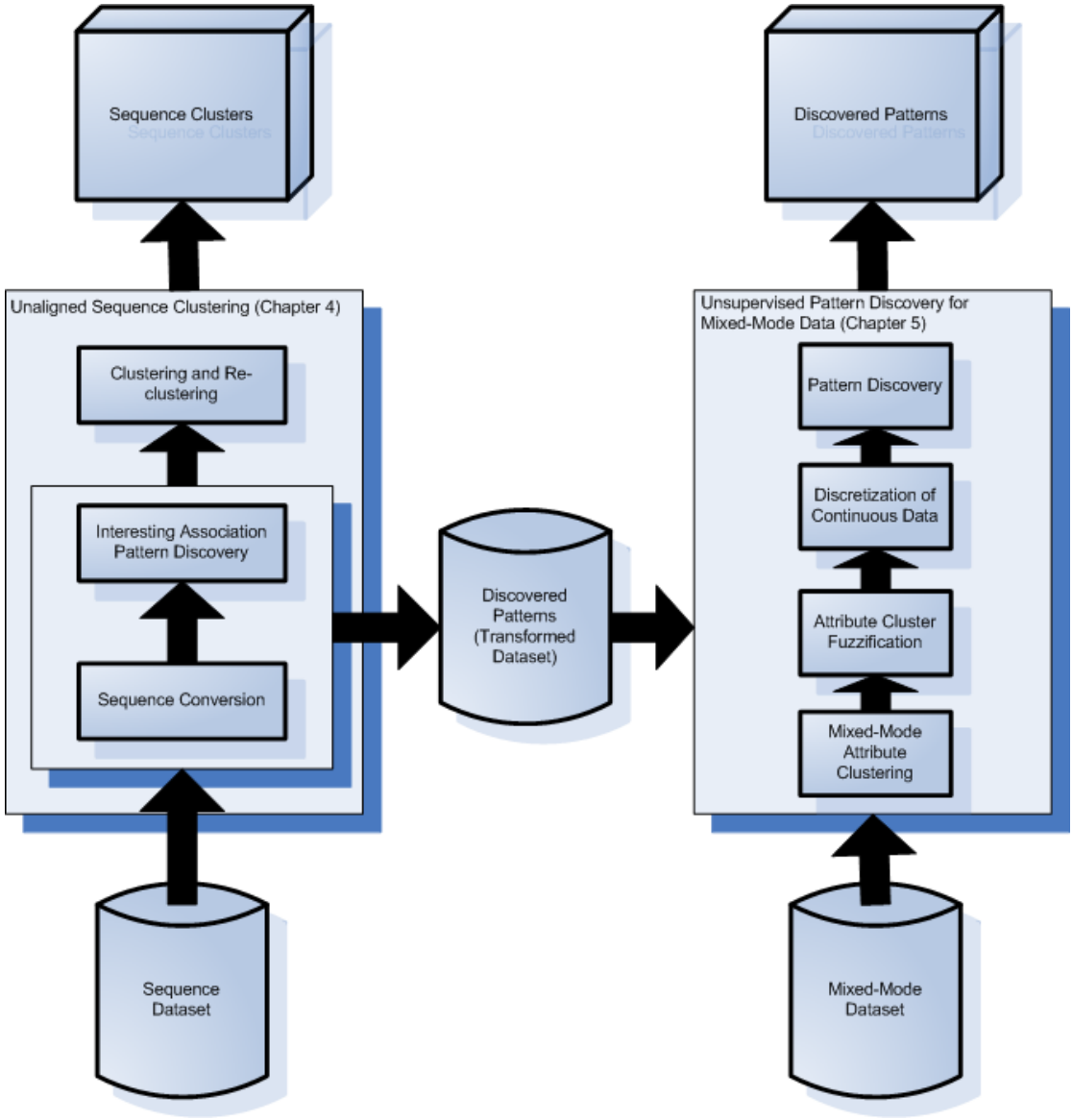


Figure 3.1. The proposed data mining approach.

The proposed data mining approach is able to deal with data of various types - sequence, symbolic, continuous, categorical, discrete and interval. A schematic diagram is shown below to illustrate the approach.

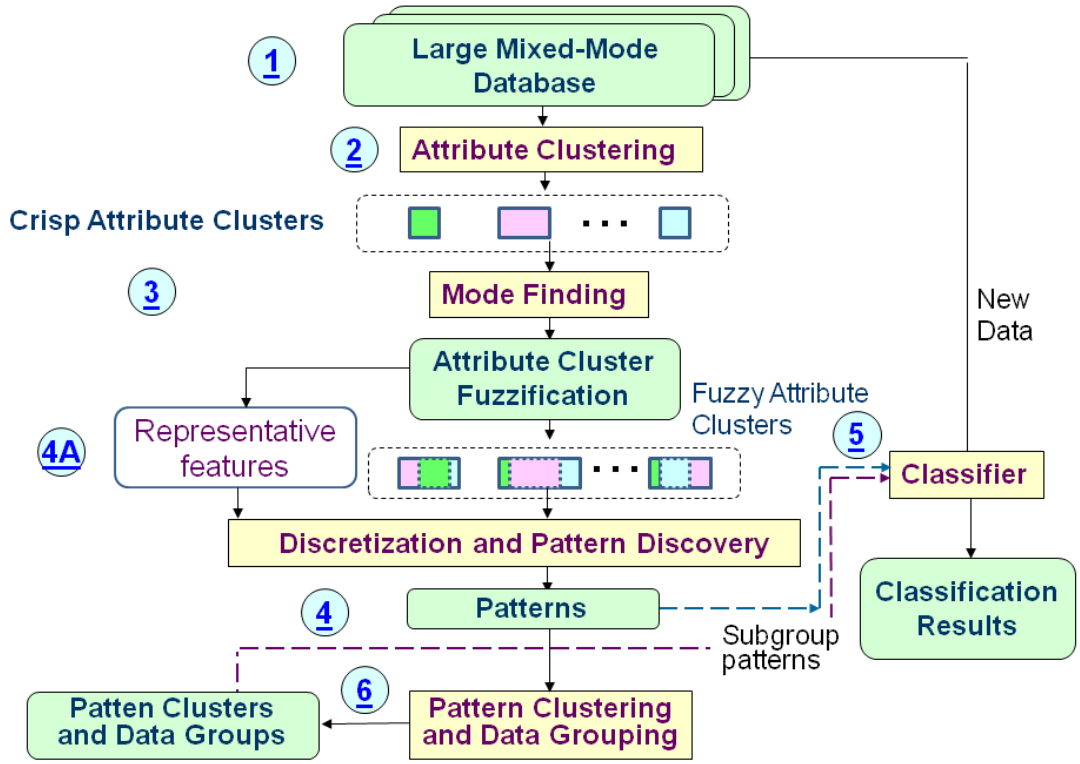


Figure 3.2. A schematic diagram for solving the problem.

The algorithms firstly perform attribute clustering to partition a database into a number of sub-databases (attribute clusters) based on the interdependence of attributes. Each attribute cluster is detected the representative attribute/feature (mode). In crisp attribute cluster case, an attribute belongs to 1 and only 1 attribute cluster. In fuzzy attribute cluster case, an attribute is able to belong to multiple attribute clusters, each with a degree of membership. Discretization is then performed to transform the continuous attributes to discretized attributes for further analysis. The discretization is conducted in every attribute cluster, each time using the mode of the attribute cluster to drive the discretization of other continuous attributes. Once all the continuous attributes are transformed into discrete attributes, pattern discovery can begin to extract patterns inherent in the data. The discovered patterns can be further clustered by pattern clustering for summarizing the numerous discovered patterns. Also, the discovered patterns can

be taken as rules to build a classifier if the class label is pull back or the clustered label is regarded as the class. When new data of various types arrive, the classifier can automatically classify/predict it into the pre-labeled class.

Chapter 4. Unaligned Sequence Clustering

Given a set of sequence data containing a number of sequences with varying length, our goal is to 1) discover interesting sequential patterns and 2) identify groups of sequences based on their sequence similarity. The proposed approach on sequence data consists of two main components as shown in Figure 4.1. In the first component, it makes use of residual analysis [Wong and Wang 1997] in statistics to test the significance of the occurrence of a sequential pattern against its expectation. The discovered patterns are then represented in the form of a relational table. It achieves the first goal. In the second component, the discovered interesting sequential patterns are then used for the clustering of sequences, adapting a two-phase clustering algorithm [Ma, Chan and Chiu 2005] which utilizes both local and global information. It tackles the second problem. Each component will be described in more detail in the following subsections. To demonstrate the effectiveness and validity of the proposed approach, extensive experiments, which contain a synthetic sequential data set, a web log sequence data set and a yeast genome data set, are conducted with comparisons to famous approaches including C5.0 decision tree, support vector machine and *k*-means clustering algorithm.

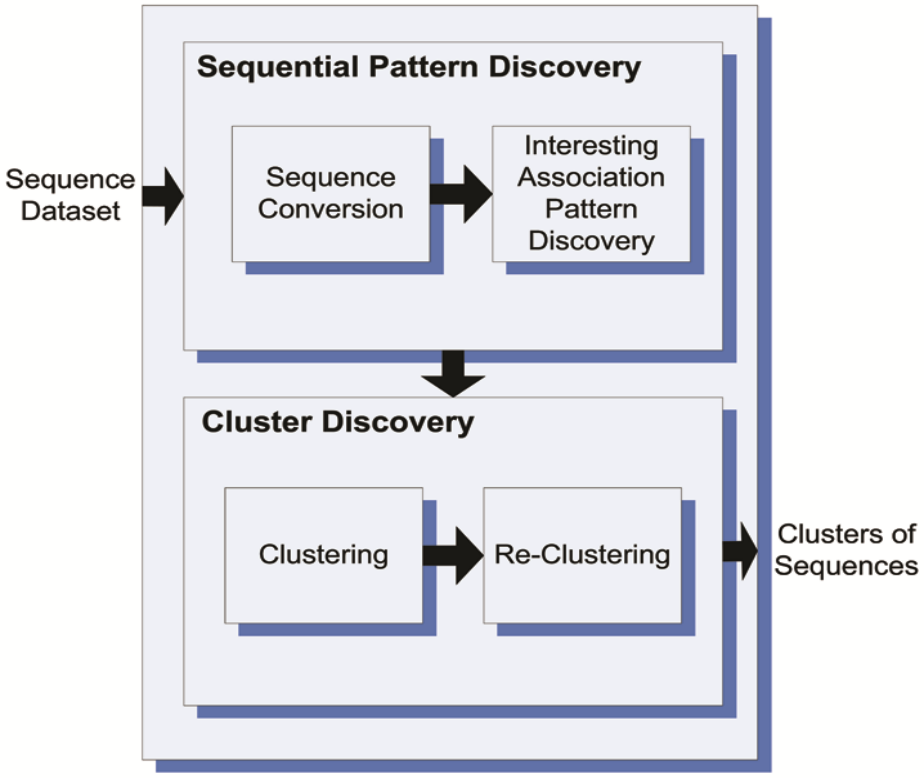


Figure 4.1. The architecture of the proposed approach on sequence data.

The rest of this chapter is organized as follows. In section 4.1, the problem of sequential pattern mining and clustering will be formally introduced. In section 4.2, the techniques to discover interesting sequential patterns and to cluster the sequences will be discussed.

4.1 The Unsupervised Sequential Pattern Mining Problem

Let us suppose that there is a set of unlabelled N sequences with varying length as shown in Table 4.1. Each sequence is represented as $S_i = \{a_{i1}, \ldots, a_{il}, \ldots, a_{iL_i}\}$, where $a_{il} \in \vartheta$ is an item, element, or ordered event, $i = 1, \ldots, N$, $l = 1, \ldots, L_i$, and ϑ is the set of alphabets that an item in a sequence can take on. Thus $\boldsymbol{L_i}$ is the length of sequence S_i. A sequence S_i with length L_i is called a L_i -sequence. A subsequence $\beta = \{b_1, b_2, \ldots, b_n\}$ of a sequence

$S_i = \{a_{i1}, \ldots, a_{il}, \ldots, a_{iL_i}\}$ is denoted as $\beta \sqsubseteq S_i$, if there exist integers $1 \leq j_1 < j_2 < \ldots < j_n \leq L_i$ such that $b_1 \subseteq a_{ij_1}$, $b_2 \subseteq a_{ij_2}$, …, $b_n \subseteq a_{ij_n}$.For example, if β = {“keyboard”, “mouse”} and S_i = {“keyboard”, “mouse”, “speaker”}, then β is a subsequence of S_i and S_i is a supersequence of β. The first goal of the problem is to discover statistically significant high-order sequential patterns. A sequence or subsequence is said to be a sequential pattern (or simply a pattern) if it is associated with one or more interestingness measures. In our proposed approach, we generalize the residual analysis [Haberman 1973] in statistics as the interestingness measure to detect the statistically significant high-order sequential patterns. A pattern with length *l* is called an l^{th} order pattern or simply an *l*-pattern. On the other hand, the discovered patterns should make the characteristics of sequences explicit. This leads to the second goal which is to partition these sequences, S_1, …, S_i, …, S_N, into clusters, $C = \{c_1, \ldots, c_k\}$, according to their sequence similarities.

Table 4.1. The definition of a sequence dataset.

Sequences	Items	Length
$\boldsymbol{S_1}$	$\boldsymbol{a_{11}}$, …, $\boldsymbol{a_{1l}}$, …, $\boldsymbol{a_{1L_1}}$	$\boldsymbol{L_1}$
⋮	…	⋮
$\boldsymbol{S_i}$	$\boldsymbol{a_{i1}}$, …, $\boldsymbol{a_{il}}$, …, $\boldsymbol{a_{iL_i}}$	$\boldsymbol{L_i}$
⋮	…	⋮
$\boldsymbol{S_N}$	$\boldsymbol{a_{N1}}$, …, $\boldsymbol{a_{Nl}}$, …, $\boldsymbol{a_{NL_N}}$	$\boldsymbol{L_N}$

4.2 The Solution to the Unsupervised Sequential Pattern Mining Problem

4.2.1 Sequence Conversion

In the first component of the proposed algorithm, it converts input sequences into subsequences. Based on the concept of sliding window, we slide a window of width *w*, which is the window size, across a sequence S_i. Then the sequence can be converted into $L_i - w + 1$ subsequences, s_{i1}, …, s_{ij}, …, $S_{i(L_i-w+1)}$, so that $s_{i1} = \{a_{i1}, a_{i2}, \ldots, a_{iw}\}$, … , $s_{ij} = \{a_{ij}, a_{i(j+1)}, \ldots, a_{i(w+j-1)}\}$, … , $S_{i(L_i-w+1)} = \{a_{i(L_i-w+1)}, a_{i(L_i-w+2)}, \ldots, a_{iL_i}\}$ [Ma and Chan 2008]. Each such a subsequence has the same length, *w*, and in our method, the item in the last position, $a_{i(w+j-1)}$, of each subsequence, s_{ij}, represents the class, C_p, this sequence belongs to in order for detecting associations between items. Thus, all subsequences are classified into one of *P* classes, C_p, $p = 1, \ldots, P$. *w* is a user input parameter, which could be decided by the rule of thumb or adjusted by experimentally assessing the performance so as to obtain the local optimum. After the conversion of each sequence into a number of subsequences, the data mining procedure can be performed.

4.2.2 Interesting Association Pattern Discovery

Interesting association relationships are discovered by detecting statistically significant associations between sequences and each class label. To begin this task, we let obs_{pq} denote the observed number of subsequences that belongs to a class, C_p, and is characterized by the same item, $a_j^{(q)}$, where $a_j^{(q)} \in \vartheta$, $j = 1, \ldots, w - 1$, $q = 1, \ldots, Q$, $Q \leq |\vartheta|$, and *Q* is the total number of

distinct items at position j of all the subsequences. Let $exp_{pq} = \frac{obs_{p+}obs_{+q}}{N'}$, where $obs_{p+} = \sum_{q=1}^{Q} obs_{pq}$ and $obs_{+q} = \sum_{p=1}^{P} obs_{pq}$ and $N' = \sum_{i=1}^{N}(L_i - w + 1)$ is the total number of subsequences formed by sliding a window of size w through all N sequences, be the expected total number of subsequences that belongs to a class, C_p, under the assumption that having a class label C_p is independent of whether or not a subsequence in this class has the characteristic $a_j^{(q)}$ [Ma and Chan 2008]. Table 4.2 shows the definition of the notations.

Table 4.2. The definition of the notations.

Item / Class	$a_j^{(1)}$	⋯	$a_j^{(q)}$	⋯	$a_j^{(Q)}$	obs_{Class+}
C_1	obs_{11}	⋯	obs_{1q}	⋯	obs_{1Q}	obs_{1+}
⋮	⋮		⋮		⋮	
C_p	obs_{p1}	⋯	obs_{pq}	⋯	obs_{pQ}	obs_{p+}
⋮	⋮		⋮		⋮	
C_Q	obs_{P1}	⋯	obs_{Pq}	⋯	obs_{PQ}	obs_{P+}
obs_{+Item}	obs_{+1}		obs_{+q}		obs_{+Q}	N'

Given obs_{pq} and exp_{pq}, an association is then determined if obs_{pq} is significantly different from exp_{pq}. We apply the adjusted residual d_{pq} [Wong and Wang 1997] to detect the association and it is defined in Equation (3.1):

$$d_{pq} = \frac{z_{pq}}{\sqrt{v_{pq}}},$$

where z_{pq} is the standardized residual [Haberman 1973] given by Equation (3.2):

$$z_{pq} = \frac{obs_{pq} - exp_{pq}}{\sqrt{exp_{pq}}},$$

and v_{pq} is the maximum likelihood estimate of its asymptotic variance [Haberman 1973] given by Equation (3.3):

$$v_{pq} = (1 - \frac{obs_{p+}}{N'})(1 - \frac{obs_{+q}}{N'}).$$

If d_{pq} > 1.96 (with 95% confidence level), an association is considered to be interesting. Therefore it is concluded that the item $a_j^{(q)}$ at position j is associated with C_p and is useful for determining if a subsequence should be classified into C_p. This association between $a_j^{(q)}$ and C_p is statistically significant and such an association is referred to an interesting pattern of order $|j|$. This interesting first-order association pattern is statistically significant between one single item and a class label.

Based on the described approach to determine a first-order pattern, it is able to determine if there is an interesting second-order association pattern involving the association between two items and a class label. The association between two items, $a_j^{(q)}$ and $a_y^{(q)}$, at positions j and y and the class label C_p is tested to detect whether or not it is statistically significant by joining the 2 items to form $a_\varphi^{(q)}$, where $\varphi = j \cup y, j, y \in \{1, \ldots, w - 1\}, j \neq y$. To avoid exhaustive search, $a_j^{(q)}$ and $a_y^{(q)}$ are chosen to join for forming $a_\varphi^{(q)}$ only if both association patterns of first-order which are $a_j^{(q)}$ and C_p, and, $a_y^{(q)}$ and C_p are interesting. In this way, the algorithm continues to search for interesting third-order association patterns involving the association between three items and a class label when all combinations of second-order patterns are interesting. Generally, this procedure tests whether or not a pattern of order *n* is interesting when all its sub-patterns of order (*n* – 1) are also interesting. Using this technique to search for higher-order

patterns can effectively avoid evaluating of all possible combinations of items exhaustively.

4.2.3 Sequential Pattern Table Construction

After all interesting patterns of different orders, $M = \{m_1, \ldots, m_{|M|}\}$, are discovered, these are used to represent the sequences in terms of a relational table, $T = \{o_{ij} \mid i = 1, \ldots, N, j = 1, \ldots, |M|\}$ where o_{ij} is the count of the occurrence of sequential pattern m_j in sequence S_i and is normalized by Equation (3.4):

$$o_{ij} = \frac{o_{ij}}{L_i - U_{m_j} + 1},$$

where L_i is the length of sequence S_i and U_{m_j} is the length of sequential pattern m_j.

Each row in the table T corresponds to one particular sequence and each column to a sequential pattern. Such a table uses the discovered sequential patterns to represent the sequences and is the output of the first component of the proposed approach on sequence data. It can be further used for data analysis, grouping and selecting the sequences and interesting patterns.

4.2.4 Clustering and Re-clustering

In the second component, the proposed approach partitions the sequences by clustering the sequential pattern table T using a two-phase clustering algorithm [Ma, Chan and Chiu 2005]. The input of this component is indeed the features discovered in the previous component. The initial phase utilizing a local pair-wise distance between two sequences groups similar sequences into clusters. The second phase is to re-cluster the data which regards the assigned sequences with cluster labels as training data to construct a classifier based on a global probabilistic measure of interestingness. The classifier which distinguishes

between relevant and irrelevant interesting patterns classifies sequences into the same cluster or a different one.

The initial phase (clustering) can be performed by any clustering algorithms. In our study, the popular clustering *k*-means clustering algorithm [MacQueen 1967] is adopted. This algorithm, when applying in clustering data in a table, iteratively assigns the rows of table T into *k* clusters where the similarity between the assigning row and the mean (centre) of the cluster is highest. The mean of the cluster is recalculated when a sequence is assigned to that cluster. For the similarity function, the Euclidean distance is used. For any two sequences, S_x and S_y, which are characterized by a set of $|M|$ sequential patterns, the measure is defined in Equation (3.5):

$$E(S_x, S_y) = \sqrt{\sum_{j=1}^{|M|}(o_{xj} - o_{yj})^2}.$$

The second phase (re-clustering) consists of a learning step and a re-evaluation step. In the learning step, the sequential pattern table T is discretized based on a popular technique Optimal Class Dependent Discretization, OCDD, which minimizes the loss of information during the process [Liu, Wong and Wang 2004]. Thus, for a particular sequential pattern m_j, the occurrence of it in all sequences, o_{1j}, …, o_{ij}, …, o_{lj}, are partitioned into intervals $o_j^{(q')}$, where q' = 1, …, Q' and Q' is the total number of distinct data intervals of m_j. After discretization, interesting association relationships are discovered in each initial cluster by detecting the associations between the occurrences of sequential patterns in sequences that belong to a particular cluster and the cluster label itself. To being this task, let $obs_{p'q'}$ denote the observed total number of sequences, S_1, …, S_i, …S_l, where $l \leq N$, in the data that belong to a given cluster, $C_{p'}$, where

p' = 1, …, k and k is the total number of initial clusters discovered, and are characterized by the occurrences of sequential patterns in sequences that are within the interval of $o_j^{(q')}$. Let $exp_{p'q'} = \frac{obs_{p'+}obs_{+q'}}{N''}$, be the expected total under the assumption that being a member of $C_{p'}$ is independent of whether or not a sequence has the characteristic $o_j^{(q')}$, where $obs_{p'+} = \sum_{q'=1}^{Q'} obs_{p'q'}$, $obs_{+q'} = \sum_{p'=1}^{k} obs_{p'q'}$, and $N'' = \sum_{p',q'} obs_{p'q'}$. An association is considered interesting if $obs_{p'q'}$ is significantly different from $exp_{p'q'}$. To determine if this is the case, the adjusted residual $d_{p'q'}$ is used. The calculation of $d_{p'q'}$ is the same as Equation (3.1) by substituting p' into p and q' into q. An association is considered to be interesting if it is statistically significant. With the adjusted residual $d_{p'q'}$, it is able to determine if $o_j^{(q')}$, of m_j, is associated with a cluster, $C_{p'}$, say a 95% confidence level ($d_{p'q'}$>1.96). If so, it can be utilized to construct a characteristic description of $C_{p'}$. This description is represented as follows. If the occurrence of m_j in a sequence is within the interval of $o_j^{(q')}$, then it is with certainty $W(Cluster = C_{p'}/Cluster \neq C_{p'}|o_j^{(q')})$ that the sequence belongs to $C_{p'}$, where W, *weight of evidence* measure [Wang and Wong 2003; Osteyee and Good 1974], is defined in terms of the mutual information $I(C_{p'}: o_j^{(q')})$ as Equation (3.6):

$$W\left(Cluster = \frac{C_{p'}}{Cluster} \neq C_{p'} \middle| o_j^{(q')}\right)$$

$$= I\left(C_{p'}: o_j^{(q')}\right) - I(\neq C_{p'}: o_j^{(q')}),$$

where

$$I\left(C_{p'}: o_j^{(q')}\right) = log \frac{\Pr\left(C_{p'}|o_j^{(q')}\right)}{\Pr\left(C_{p'}\right)}.$$

Weight of evidence measures the amount of positive or negative evidence that is provided by characteristic $o_j^{(q')}$, supporting or refuting the labeling of a sequence as $C_{p'}$. It is a probabilistic interestingness measure, being effective to deal with incomplete, missing or erroneous data. If the characteristic is relevant in determining the cluster membership, it is reflected by the interestingness measure.

In the re-evaluation step, the cluster membership of a sequence, S_i, characterized by $m_1, \ldots, m_j, \ldots, m_{|M|}$, can be matched against the discovered associations. If the occurrence of the sequential pattern, o_{ij}, of S_i satisfies the associations (i.e., o_{ij} of S_i is within the interval of $o_j^{(q')}$) that implies $C_{p'}$, then we can conclude that the description of S_i partially matches that of $C_{p'}$. By repeating the procedure, which is by matching each o_{ij}, $j = 1, \ldots, |M|$, of S_i against the discovered associations, total weight of evidence of assigning S_i to $C_{p'}$ can be computed. Suppose that of the $|M|$ characteristics which describe S_i, only β, $\beta \leq |M|$, of them are found to match with the discovered associations. Then, *total weight of evidence* supporting the labeling of S_i as $C_{p'}$ is defined in Equation (3.7):

$$TotalW(Cluster = C_{p'}/Cluster \neq C_{p'}|o_1^{(q')}, \ldots, o_j^{(q')}, \ldots, o_\beta^{(q')})$$

$$= \sum_{j=1}^{\beta} W(Cluster = C_{p'}/Cluster \neq C_{p'}|o_j^{(q')}).$$

Then S_i is assigned to $C_{p'}$ if:

$$\max_{p' \in \{1, \ldots, k\}} \left\{ \sum_{j=1}^{\beta} W(Cluster = C_{p'}/Cluster \neq C_{p'}|o_j^{(q')} \right\}.$$

The re-clustering phase described above allows for probabilistic associations to be detected. It is done by distinguishing between relevant and

irrelevant occurrences of sequential patterns in sequences and taking into consideration global information contained in a specific cluster arrangement by evaluating the importance of different occurrences of sequential patterns in sequences in determining cluster memberships. This feature makes the proposed algorithm more robust in treating noisy data compared to those algorithms relying only on local pair-wise similarity measures.

4.3 Experiments and Results

In order to evaluate the performance of the proposed method, experiments are conducted on a synthetic dataset, then on a well-known msnbc.com anonymous web log dataset, and on a yeast genome sequence dataset. Here we indicate that the novelty of the proposed method is not the clustering method, but a unified framework for the discovery of appropriate pattern information from sequence data which considers both local and global characteristics of the data.

4.3.1 Synthetic Dataset

To evaluate the clusters and patterns of sequences formed by the proposed method, we first applied it to a synthetic dataset. Each sequence in the synthetic dataset is composed of an arbitrary number of items, from 50 items to 100 items, by a pseudorandom number generator and is preclassified into one of the three classes: C_1, C_2, and C_3. Each item can take on one of the three alphabets: A, B and C. In the designed experiment, each sequence is implanted a sequential pattern which can determine the class membership. This class membership is taken as the ground truth for the problem. We would like to see if the proposed method can detect high-order patterns and cluster sequences according to their pattern similarities. The sequential patterns A*AAA, B*B*B, and CCCC are

statistically significant patterns which belong to C_1, C_2, and C_3, respectively, where * is a special alphabet, i.e. "don't care" symbol, simply meaning that it can be any alphabet of A, B or C. All these artificially implanted statistically significant sequential patterns are put into sequences in random positions. For a clustering algorithm of unaligned sequential data to be effective, it should be able to reveal such patterns and group sequences exhibiting similar ones into the same cluster. In our experiment, we generated 15,000 sequences in the synthetic dataset and each class contains 5,000 sequences. Noises were added to the dataset by replacing the sequential patterns in 25 percent of the sequences with alphabets randomly drawn from A, B or C.

For the purpose of comparison, we calculate *Recall* and *Precision* values of the clustering result to the synthetic dataset. *Recall* specifies the probability of correctly predicting a classifier and it is defined in Equation (3.8):

$$Recall = TP / (TP + FN),$$

and *Precision* specifies the probability that the provided prediction is correct and it is defined in Equation (3.9):

$$Precision = TP / (TP + FP),$$

where *TP* (True Positives) is the number of correctly identified true pairs, *FN* (False Negative) is the number of not identified true pairs and *FP* (False Positive) is the number of false pairs predicted to be true pairs. A pair of sequence is considered to be a true pair if both are in the same cluster. *F-measure* combines the *precision* and *recall* values. The *F-measure* is defined in Equation (3.10):

$$F\text{-}measure = \frac{(1+\beta^2)\cdot Precision\cdot Recall}{(Precision+Recall)}.$$

In (3.10), β is a positive constant that specifies the relative importance of *Precision* and *Recall* values. When $\beta = 1$, the importance of precision and recall

values is equal and *F-measure* is the harmonic mean of them. We set $\beta = 1$ in our experiment. A higher *F-measure* implies a better quality of the clustering result where *F-measure* = 1 means the perfect cluster.

To evaluate the performance, we first discovered interesting sequential patterns from the synthetic dataset by the first component of the proposed method and investigated the output of it before the second component began. This experiment illustrates how it reveals high order sequential patterns in the initial stage. We set the sliding window size *w* to 5 here since in the synthetic dataset, the length of artificially embedded patterns is 5. The first component outputs the sequential pattern table which contains 37 interesting sequential patterns, including 11 first-order patterns, 15 second-order patterns, 9 third-order patterns and 2 fourth-order patterns. Among the interesting sequential patterns, all previously implanted patterns (A*AAA, B*B*B and CCCC) are identified and ranked top 3 strong patterns according to their adjusted residue values. In the second component of proposed method, we then applied *k*-means algorithm [MacQueen 1967] and, afterwards, the two-phase clustering algorithm to the sequential pattern table. The optimal number of cluster *k* was found to be 3 in *k*-means algorithm by comparing the *F-measure* of different clustering results generated by different values of *k*. After a number of experimentations, we selected the best clustering result obtained by *k*-means algorithm and put it into the second phase clustering. As reported by the experiment, the *Recall, Precision* and *F-measure* values of the clustering result generated by *k*-means algorithm is 76.65%, 69.76% and 73.04% respectively and these by the two- phase clustering algorithm are 91.98%, 83.71% and 87.38% respectively. It is clear that the sequential pattern table generated by the first component of the proposed method

enables *k*-means algorithm (phase 1 of the second component) to produce acceptable clustering results. We attribute this to the residual analysis in searching for interesting patterns of sequences. It also shows that the addition of the two-phase clustering algorithm in the framework boosts the clustering quality eventually. Overall, the proposed approach can identify useful patterns from a collection of discovered patterns and these useful patterns are used to reclassify the sequences into clusters by weight of evidence measure. Weight of evidence measure has the capability of taking partial information from useful sequential patterns into account and combining the global information of them to identify non-linear patterns.

4.3.2 Web Log Dataset

The msnbc.com anonymous web data, a real-life dataset, is obtained from Internet Information Server (IIS) logs for msnbc.com through UCI site of machine learning archive [Asuncion and Newman 2007]. This dataset describes the page visits of users who visited msnbc.com on September 28, 1999. Each sequence in the dataset corresponds to page views of a user during that twenty-four hour period. Each event in the sequence corresponds to a user's request for one of the 17 page categories, including frontpage, news, tech, local, opinion, on-air, misc, weather, health, living, business, sports, summary, bbs (bulletin board service), travel, msn-news, and msn-sports. The number of sequences is 989,818 with an average number of visited pages per user (an average length of a sequence) 5.7. This dataset has been used by others [Dias and Cortinhal 2008; Cadez et al. 2003; Dias and Vermunt 2007] to extract interesting browsing patterns to study the user browsing behaviors.

In our study, we selected samples with different length to create 6 sampled datasets to perform a number of experiments to justify the effect of different window sizes in the proposed algorithm. We set the number of clusters to 2 consistent with [Dias and Cortinhal 2008; Cadez et al. 2003; Dias and Vermunt 2007]. Specifically, in our experiments, we tried using different window sizes (i.e. w = 2, 3, 5, 10, 15, 20) to the corresponding sampled datasets. The sampled datasets are generated in the following manner. For an experiment with window size w, the users with a sequence of length lesser than or equal to $w - 1$ were filtered so that in the sampled datasets, each user is associated with a sequence of length at least w.

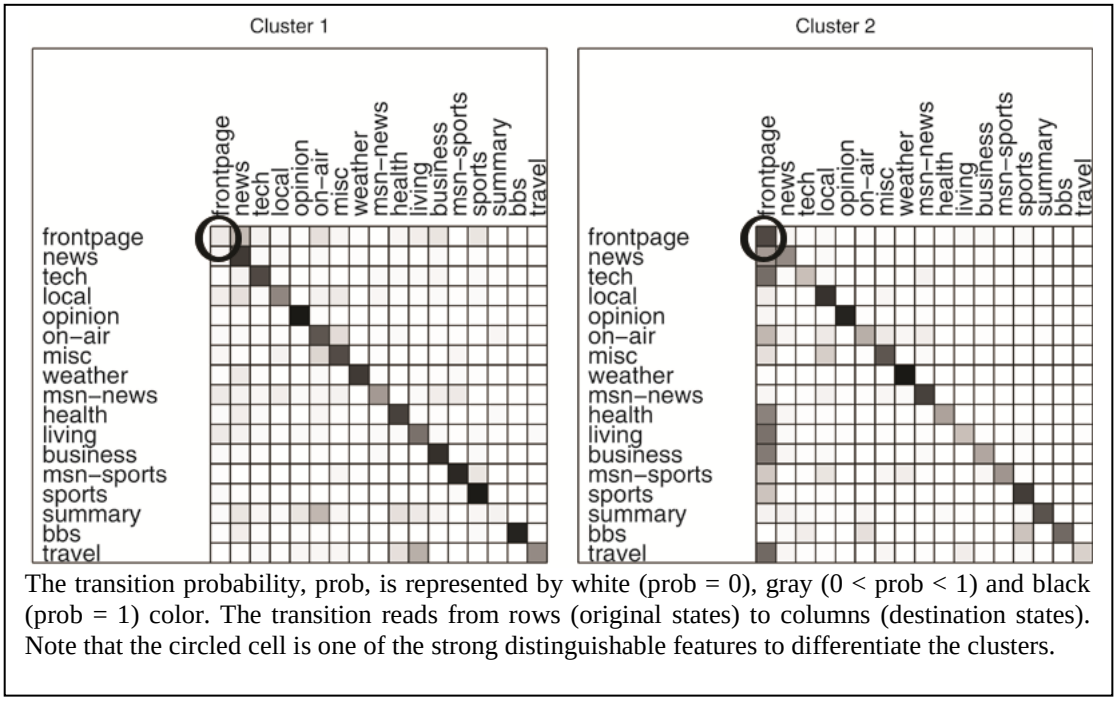
Table 4.3. Window sizes, filtered length of sequences, sample sizes, cluster sizes and number of patterns.

WS	FL	SS	C1 (%)	C2 (%)	P #
2	1	704,747	69.27	30.73	48
3	2	524,948	79.53	20.47	103
5	4	363,358	73.47	26.5	187
10	9	169,051	65.35	34.65	456
15	14	88,592	67.75	32.25	522
20	19	52,347	64.44	35.56	636

Key: WS – Window size. FL – Filtered length. SS – Sample size. C1 – Cluster 1. C2 – Cluster 2. P # – Number of patterns.

Table 4.3 provides a summary of the best results obtained by the proposed algorithm. Cluster 1 is the largest cluster with an average 69.97% of users while cluster 2 is with 30.03%. The number of patterns increases when the window sizes increases due to more high order patterns discovered. Among all

sampled datasets, there are some distinguishable sequential patterns to separate users into 2 groups. For instance, users in cluster 1 have no pattern of browsing frontpage for 2 times consecutively while all users in cluster 2 exhibit this browsing pattern. This founding is consistent with those reported by [Dias and Cortinhal 2008]. In the transition matrix as shown in Figure 4.2, it is obvious that the transition probability of "frontpage to frontpage" is very high in cluster 2 while it is close to 0 in cluster 1. There are also other discovered patterns which conform to those discussed in [Dias and Cortinhal 2008; Dias and Vermunt 2007]. In addition to first-order patterns, the proposed algorithm also discovered other high-order browsing patterns which are potentially important to understand more about the user browsing behaviors. These patterns are such as [frontpage → * → * → news], [frontpage → * → * → frontpage], [news → * → * → * → summary] and [news → * → * → * → tech], where * is a "don't care" symbol, meaning that could be any of pages.



The transition probability, prob, is represented by white (prob = 0), gray (0 < prob < 1) and black (prob = 1) color. The transition reads from rows (original states) to columns (destination states). Note that the circled cell is one of the strong distinguishable features to differentiate the clusters.

Figure 4.2. Transition matrix within each cluster [Dias and Cortinhal 2008].

To examine our approach in terms of the quality of clusters, for each clustering result of a sampled dataset, we randomly selected 30% of labeled samples as training data to build classification models. The cluster label is regarded as the class label. For building classification models, all 6 sampled datasets were fed into C5.0 (a commercial version of C4.5 [Quinlan 1993]) and mySVM (a Java implementation of Support Vector Machine [Ruping 2000]), which are both popular classification algorithms. After building the classification models, the 6 sampled datasets were unlabelled and were put into the models to obtain the classification accuracy. Figure 4.3 summarizes the classification results. Among 6 sampled datasets, all 2 trained classification models attain satisfactory results with an average accuracy of 94.6%. The experimental results demonstrates the usefulness of the proposed algorithm in real world data analysis that it is a practical and effective technique to group web users into clusters based on their browsing patterns' similarities and to allow classification algorithms to build accurate classification models.

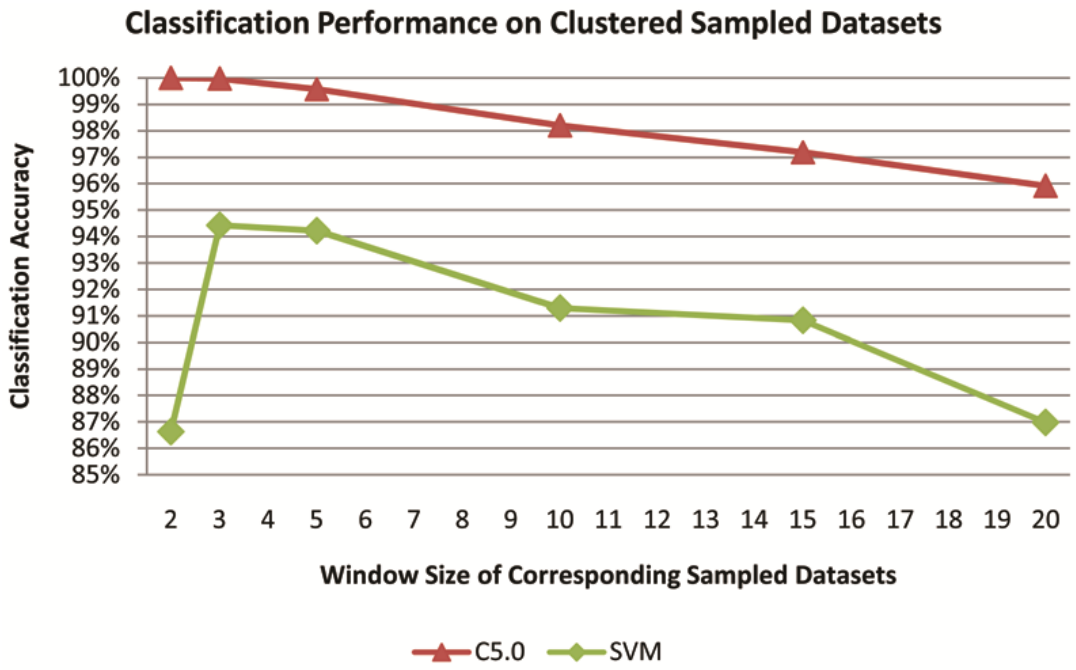


Figure 4.3. The plot of classification accuracy values against *w*, window size of sampled dataset.

4.3.3 Yeast Genome Sequence Dataset

This experiment is to examine the capability of the proposed algorithm in discovering biological functional units. In gene expression regulations, understanding the genome sequence is the fundamental step to understand the complex mechanism of gene expression. The proposed algorithm is applied on a set of yeast genome sequence data in order to identify the functional and regulatory groupings governed by the transcription factor binding sites. The reason to choose the yeast dataset, especially Saccharomyces cerevisiae, is due to its well studies and known transcription factors along with their regulated genes. The dataset is available in SCPD database, the Promoter Database of Saccharomyces cerevisiae [SCPD 2010]. The binding sites for the DNA sequences are determined experimentally in the database. This genome sequence

dataset is composed of the regulated genes from the upstream (promoter) regions. Each DNA sequence is associated with one or more transcription factors. The genes are believed to be co-regulated by specific transcription factors. A set of genes associated with the same transcription factor is called a regulon. In our dataset, there are totally 109 yeast genome sequence classified into 18 regulons including CAR1, CPF1, CSRE, GCN4, GCR1, MATalpha2, MCB, MIG1, PDR3, PHO4, RAP1, REB1, ROX1, SCB, SFF, STE12, TBP and UASPHR. Among the regulons, each of them has at least 3 genes with all consensus binding sites available. For each regulon of the transcription factors, DNA sequences are extracted from the upstream (promoter) regions from position -800 to +1 that is relative to the ORF (translation start site) so that all sequences have the same length of 801.

In our experiments, we input the 109 sequences into the proposed sequence clustering algorithm. These sequences are labeled and are classified into 18 regulons so we take this class information as the ground truth of the problem. We hide the class labels before the analysis. To study the sensitivity, we set window sizes from 2 to 10 in this set of experiments. In our proposed algorithm, it first detects the sequential patterns from the yeast genome sequences and then uses the discovered patterns for clustering. Table 4.4 and Figure 4.4 shows the experimental results.

Table 4.4. Performance of the propose algorithm on regulons of yeast genome sequences from SCPD.

Window Size	Discovered Pattern #	Clustering Accuracy	
		k-means	Proposed
2	5	84.08%	86.92%
3	18	82.72%	86.00%
4	56	84.05%	86.15%
5	145	82.08%	85.85%
6	241	83.47%	85.73%
7	492	78.34%	85.56%
8	902	76.16%	85.24%
9	1878	77.81%	83.81%
10	4171	57.76%	84.15%

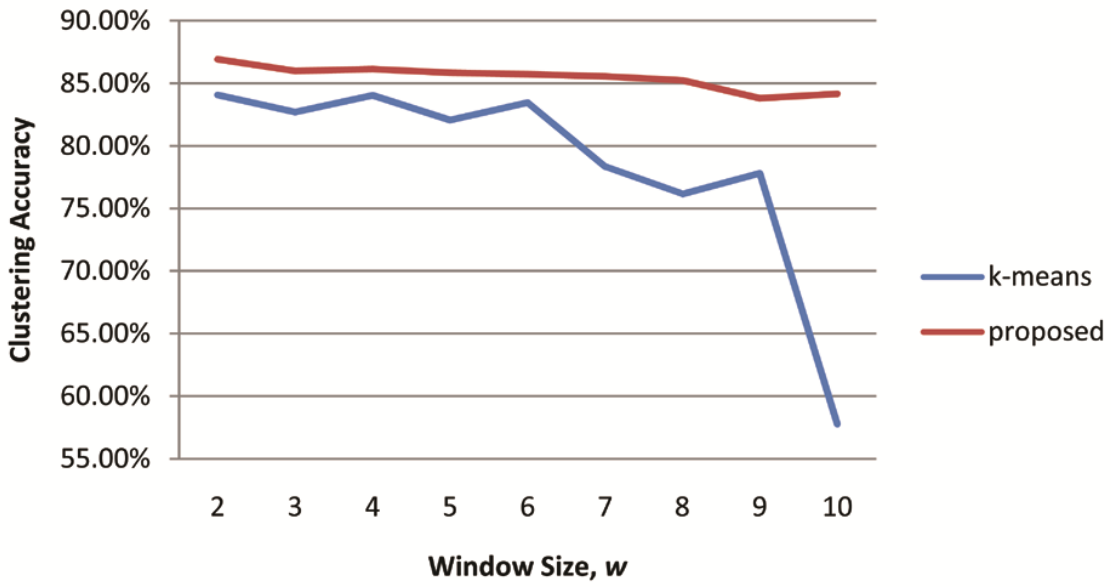


Figure 4.4. The plot of clustering accuracy values against w, window size of yeast genome sequences.

From the experimental results, it is reported that the discovered patterns do effectively describe the characteristics (transcription factor binding sites) of the yeast genome sequences for the cluster analysis to separate the sequences into 18 regulons based on their characteristics. As expected, the number of patterns increases as the window size increases. It is interesting to note that for window size from 2 to 9, both *k*-means clustering and the proposed two-phase clustering achieve high rates of accuracy. This conforms to the fact that in DNA sequences, tandem repeats are common. For instance, in a sequence "GGGAAAAAAA", the pattern "AAAA" occurs at position 4, 5, 6 and 7 which overlap multiple times. In setting large window size, many those patterns will be discovered and further filtering step needs to be taken. To avoid these, using small window size (i.e. 2, 3, and 4) is already a possible way which is fair enough to extract the information to characterize the sequences. The effect of it is obvious as shown in Figure 4.4, small window size facilitates good clustering performance. Nevertheless, the proposed two-phase clustering is able to handle noisy information by distinguish between relevant and irrelevant patterns even though large window size leads to duplicated patterns while *k*-means is not (see the case when window size is set to 10 with 4,171 discovered patterns). It is attributed to the weight of evidence measure in the proposed algorithm that irrelevant information (patterns) are given lesser weights in supporting the labeling of a sequence to the correct cluster and relevant information are concerned.

4.4 Summary

In this chapter, we have introduced the proposed approach that supports the discovery of useful sequential patterns and clusters in a sequence database. Its capability has been demonstrated by 3 sets of experiments with large datasets.

It is first applied to a synthetic dataset to verify its effectiveness. Then applying it to a web log dataset to discover patterns of browsed pages that could be used to characterize web users and distinguish them from each other as well as group them together. The experimental results using a synthetic dataset and a real-life web log dataset show that the proposed approach produces meaningful clustering results and makes the hidden interesting sequential patterns explicit. Applying it to a real-life web log dataset, we found that groups of web users exhibit similar browsing patterns and how these patterns supports or refutes the labeling of a particular web user to a group. This work as illustrated in the experiment is focused on effective clustering of web users based on detected browsing patterns. The results from the yeast transcription factor experiments demonstrate that relevant functional information can be extracted and further calibrated for meaningful clustering of regulated genes. All these results confirm the algorithm has the ability to acquire interesting and unknown information inherent in the sequences. The proposed algorithm can be further applied as a generic sequence data analysis tool in many application domains.

Chapter 5. Unsupervised Pattern Discovery for Mixed-Mode Data

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

Consider a data set D containing a set of N-tuples of mixed-mode data. Every tuple is described by N attributes, some assuming discrete values and others continuous values. Let $\boldsymbol{X} = \{X_1, \ldots, X_N\}$ represent this attribute set. For convenience, let us permute the attributes (without affecting the analysis) that the first M attributes $\{ Xi | 1 \leq i \leq M\}$ are discrete valued and the remaining $\{Xi' | M+1 \leq i' \leq N\}$ are continuous valued. Then, each X_i, $1 \leq i \leq M$ can be seen as a discrete random variable taking on values from its alphabet $\alpha_i = \{\alpha_i^1, \ldots, \alpha_i^{m_i}\}$, where m_i is the cardinality of the alphabet of the i^{th} attribute. Each X_i, $M+1 \leq i \leq N$ can then be seen as a continuous random variable. Thus, a realization of $\boldsymbol{X}$ can be denoted by $x_k = \{x_{1k}, \ldots x_{ik} \ldots x_{Mk}, x_{(M+1)k}, \ldots x_{i'k} \ldots x_{Nk}\}$ where $\{ x_{ik} | 1 \leq i \leq M\}$ can assume any value in α_i and $\{ x_{i'k} | M+1 \leq i' \leq N\}$ can assume any value in $\{ M_{i'k} \leq \Re \leq N_{i'k}\}$ where $\Re$ is the real number. Thus, each tuple in the data set is a realization of $\boldsymbol{X}$.

In this chapter, we will address three problems: 1) whether the dataset contains attributes which characterize different subgroups within the attribute set; 2) whether the dataset contains various attribute subsets, each of which contain subgroups characterized by their attributes; 3) whether the dataset contains attributes which may have strong correlation to more than one subgroup, or may

associate with patterns which might overlap different subgroups. Among these problems, class labels are not available in the dataset or not used in the analysis. We will propose a new approach to tackle them.

To handle problem 1) and 2), attribute clustering is conducted to obtain attribute subgroups (clusters) so that attribute within an attribute cluster should have high correlation with or high interdependence to each other, whereas attributes in different attribute clusters are less correlated or more independent. We then identify the most representative attribute (referred to as the mode) in each attribute cluster as one with strongest interdependence with all other attributes in the subgroup. To handle problem 3), we fuzzify each of the crisp attribute clusters by drawing in attributes which share the fuzzy membership into the original crisp clusters. Unlike others' work in data mining / pattern discovery, our work is dealing with attributes which could take on categorical (discrete) and/or continuous values, which is a mixed-mode space.

As mentioned in the previous chapters, 2 major inter-related challenges in the current pattern discovery algorithms on mixed-mode databases are, firstly, large attribute size and, secondly, the discretization of the continuous data. The first challenge is tackled when mixed-mode attribute clustering proposed in this chapter is applied. To tackle the second challenge effectively, class dependent discretization algorithms, which maximize the interdependence between the interval values derived from the discretization of the continuous attributes and the given class labels [Liu, Wong and Wang 2004; Ching, Wong and Chan 1995], could be applied. For years, most effective classification algorithms in machine learning can only be applied to nominal (categorical) database or database with continuous values separately [Wong and Wang 1997] but are unable to deal with

mixed-mode database directly. Recently, researchers also found that even if some systems are explicitly designed for continuous attributes, they can attain a higher accuracy if continuous data are appropriately discretized. Through discretizing the continuous data, most of the inductive learning algorithms can accommodate continuous and mixed-mode data more effectively [Wang and Wong 2003; Chau and Wong 1999; Chiu, Wong and Cheung 1991]. However, in our problem, the class labels are not available and in fact, most real world problems are without prior knowledge. Therefore, the concept of maximizing class-attribute dependence is not readily and easily applied for discretizing the continuous data space. To solve this, we take the mode of each attribute cluster functioning like the class label to drive the discretization. We will adopt an optimum iterative dynamic programming algorithm known as OCDD (Optimal Class-Dependent Discretization Algorithm) [Liu, Wong and Wang 2004] for discretization once the mode or representative attribute is chosen for an attribute cluster. Such a process could be viewed as partitioning of the outcome values of a continuous-valued attribute into a number of discrete intervals that maximize its interdependence with the mode.

Once a mixed-mode database is transformed into one containing only categorical events, pattern discovery [Wong and Wang 2003] methodology could be readily applied to the transformed database to constitute a unified pattern discovery framework such that all the definitions for events, event associations and patterns will be based on discrete variables.

Our proposed methodology is composed of 4 phases: mixed mode attribute clustering, attribute cluster fuzzification, continuous data discretization and discovery of statistically significant patterns.

To demonstrate the effectiveness and validity of the proposed approach, extensive experiments, which contain 2 sets of synthetic mixed mode data, a collection of data sets from UCI machine learning archive and 2 real world large data sets, are conducted with comparisons to famous approaches including C5.0 decision tree, optimal class dependent discretization (OCDD) algorithm and k-modes attribute clustering algorithm.

5.1 Mixed-Mode Attribute Clustering

Mixed-Mode Attribute Clustering Algorithm (MACA) [Wong et al. 2010] to cluster attributes based on the interdependence among their attribute values is evolved from the Attribute Clustering Algorithm (ACA) [Au et al. 2005] which requires continuous valued data to be discretized using class information. Unlike ACA, MACA can effectively operate without data discretization and class information. Meaningful attribute clusters (groups) could be found by MACA such that attributes within an attribute cluster have high interdependence with each other, whereas attributes in different attribute clusters are less correlated. MACA uses a normalized interdependence redundancy measure

$$R(A_i: A_j) = \frac{I(A_i:A_j)}{H(A_i:A_j)}, \qquad (5.1)$$

to account for interdependence between attributes where $I(A_i: A_j)$ is the mutual information between A_i and A_j , and $H(A_i: A_j)$ is the joint entropy of A_i and A_j.

Let us denote σas the SELECT operation from relational algebra and $|S|$ as the cardinality of set *S*. To calculate $I(A_i: A_j)$ and $H(A_i: A_j)$ between discrete-valued data, we first define the probability of a tuple in *D* having $A_i = \alpha_i^k, i \in \{1, \dots, M\}, k \in \{1, \dots, m_i\}$ as:

$$\Pr\left(A_i = \alpha_i^k\right) = \frac{\left|\sigma_{A_i=\alpha_i^k}(D)\right|}{\left|\sigma_{A_i\neq \mathrm{NULL}}(D)\right|}$$

and the joint probability of a record in D having $A_i = \alpha_i^k$ and $A_j = \alpha_j^l$, $i, j \in \{1, \ldots, M\}, i \neq j, k \in \{1, \ldots, m_i\}, l \in \{1, \ldots, m_j\}$ as:

$$\Pr\left(A_i = \alpha_i^k \wedge A_j = \alpha_j^l\right) = \frac{\left|\sigma_{A_i=\alpha_i^k \wedge A_j=\alpha_j^l}(D)\right|}{\left|\sigma_{A_i\neq \mathrm{NULL} \wedge A_j\neq \mathrm{NULL}}(D)\right|}.$$

Once we have $\Pr\left(A_i = \alpha_i^k\right)$ and $\Pr\left(A_i = \alpha_i^k \wedge A_j = \alpha_j^l\right)$ defined, we can define $I(A_i: A_j)$ and $H(A_i: A_j)$. $I(A_i: A_j)$ is defined as:

$$I\left(A_i: A_j\right) = \sum_{k=1}^{m_i}\sum_{l=1}^{m_j} \Pr\left(A_i = \alpha_i^k \wedge A_j = \alpha_j^l\right) \log \frac{\Pr\left(A_i = \alpha_i^k \wedge A_j = \alpha_j^l\right)}{\Pr\left(A_i = \alpha_i^k\right)\Pr\left(A_j = \alpha_j^l\right)}$$

and $H(A_i: A_j)$ is defined as:

$$H\left(A_i: A_j\right) = -\sum_{k=1}^{m_i}\sum_{l=1}^{m_j} \Pr\left(A_i = \alpha_i^k \wedge A_j = \alpha_j^l\right) \log \Pr\left(A_i = \alpha_i^k \wedge A_j = \alpha_j^l\right).$$

$I\left(A_i: A_j\right)$ measures the average reduction in uncertainty about A_i that results from learning the value of A_j. Since its value increases with the number of attribute values, $I\left(A_i: A_j\right)$ should be normalized by $H\left(A_i: A_j\right)$, yielding the interdependence redundancy measure $R\left(A_i: A_j\right)$. $R\left(A_i: A_j\right)$ reflects the degree of deviation from independence between A_i and A_j. If $R\left(A_i: A_j\right) = 1$, A_i and A_j are strictly dependent. If $R\left(A_i: A_j\right) = 0$, A_i and A_j are statistically independent. If $0 < R\left(A_i: A_j\right) < 1$, A_i and A_j are partially dependent. If two attributes are dependent on each other, they are more correlated with each other when compared to two independent attributes. Therefore, it is able to use it to detect the interdependence or correlation of attributes. If $R\left(A_i: A_j\right) > R\left(A_i: A_q\right), q \in \{1, \ldots, M\}, q \neq i \neq j$, the dependence between A_i and A_j is greater than that between A_i and A_h.

To compute *R* between continuous valued data, we use a contingency table with as many bins as possible. Let |*T*| be the sample size of the dataset, *m* be the number of bins and α be the least number of data points in a cell. In practice, α is the parameter chosen in the rule of thumb manner (say 2 or 3), ensuring that each cell in the contingency table will have at least α data points. Thus, the number of bins is calculated as:

$$m \leq \sqrt{\frac{|T|}{\alpha}}.$$

Once m is set for A_i , $1 \leq i \leq N$, the attribute values can be treated as discrete valued attributes and *I*, *H* and *R* can be computed for attribute clustering.

To compute *R* between a discrete attribute and a continuous attribute, we first use the discrete attribute to drive the discretization of the continuous attribute by OCDD. Then the discretized continuous attribute is treated as discrete valued attributes and *I*, *H* and *R* can be computed then. If we would like to discretize the continuous attribute without considering the dependence between it and the discrete attribute, we could use entropy maximization [Wong and Chiu 1987] to discretize the continuous attribute before computing *I*, *H* and *R* for attribute clustering.

MACA is based on the *k*-mode attribute clustering algorithm that finds disjoint attribute clusters. Evolved from the *k*-means algorithm, it uses the mode instead of the means for samples to represent the center of the attribute cluster and clusters attributes instead of samples using interdependence redundancy measure *R* between attributes instead of the Euclidean distance between samples in the *k*-means algorithm. The mode denoted by η_r is the most representative attribute in cluster *r* found by:

$$MR(A_i) \geq MR(A_j) \text{ for all } j \in \{1, \dots, p\},\ i \neq j,$$

where

$$MR(A_i) = \sum_{j=1}^{p} R(A_i : A_j)$$

is the multiple interdependence redundancy measure [Au et al. 2005] of A_i within the attribute cluster r with p attributes.

In MACA, we use the *k*-mode attribute clustering algorithm to obtain k clusters iteratively until the sum of the multiple interdependence redundancy measure [Au et al. 2005] of all the representative attributes denoted by *SR* is maximized. *SR* is defined as:

$$SR = \sum_{r=1}^{k} \sum_{A_i \epsilon C_r} R(A_i : \eta_r)\,.$$

MACA then determines the optimal number of clusters by optimizing the intra-group attribute interdependence over different *k*. *k* is selected such that

$$k = argmax_{k \epsilon \{2, \dots, N\}} \sum_{r=1}^{k} \sum_{A_i \epsilon C_r} R(A_i : \eta_r)\,.$$

The output yields a local optimal configuration of attribute clusters each of which contains a mode.

5.2 Attribute Cluster Fuzzification

Now from the attribute clusters obtained, each cluster $C_r \epsilon \{1, \dots, k\}$ contains a mode η_r and every attribute $A_i, i \in \{1, \dots, N\}$, is assigned to only 1 attribute cluster $C_r \in \{C_1, \dots, C_k\}$, where the attribute clusters are disjoint, i.e. $C_r \cap C_s = \emptyset$ for all $s \in \{1, \dots, k\} - \{r\}$. However, if situations arise that an attribute may have strong correlation to more than one attribute cluster, or may associate with patterns which might overlap different attribute clusters, they may

not be found by our method at this stage. Hence we move to the second phase of our method to fuzzify the crisp attribute clusters obtained by assigning attributes to multiple attribute clusters with varying degrees of fuzzy membership such that overlapping relationship such as high order patterns among disjoint attribute clusters could be considered. This extends Mixed-Mode Attribute Clustering Algorithm (MACA) to Fuzzy Mixed-Mode Attribute Clustering Algorithm (FMACA).

To construct the fuzzy membership, the interdependence redundancy measure *R* (Equation (5.1)) is used to derive a fuzzy interdependence redundancy. Given that each attribute is with a certain *R* value to the mode of each attribute cluster, we calculate a degree of fuzzy membership of an attribute as the fractional part of the total possible membership assigned to the current attribute cluster. It is defined as below.

$$\mu_r(A_i) = \frac{1}{\sum_{c=1}^{k} \left(\frac{R(A_i:\eta_c)}{R(A_i:\eta_r)}\right)^{\frac{2}{m-1}}}$$

is the fuzzy membership function that returns the degree of membership of attribute *i* in attribute cluster *r*, where *k* is the optimal number of attribute clusters, m is the fuzzification parameter, $R(A_i:\eta_c)$ is the interdependence redundancy between attribute i and the mode of attribute cluster *c*, and $R(A_i:\eta_r)$ is the interdependence redundancy between attribute *i* and the mode of attribute cluster *r*. It has been shown that the following property (Equation 5.7) is desirable for the stability of fuzzy logic controllers [Pedrycz and Gomide 1998; Yen and Langari 1999].

$$\sum_{r=1}^{k} \mu_r(A_i) = 1; \qquad i = 1, 2, 3, \ldots, N$$

The fuzzification parameter m is a real number > 1 for normalizing and fuzzifying the measure. For m close to 1, the attribute closest to the mode (representative attribute) is given more weight than others. The bigger the m is the fuzzier the membership values of the attributes are.

With the fuzzy membership function defined, we can consider the correlation of each attribute with different attribute clusters among the entire data space.

5.3 Discretization of Continuous Data

To make use of the information extracted by attribute clustering and attribute fuzzification for classification and visualization, this phase involves discretizing the domains of continuous attribute values into interval events by maximizing the interdependence between the continuous attribute values and the mode (the representative attribute) using Optimal Class-Dependence Discretization (OCDD) [Liu, Wong and Wang 2004].

For completeness of the thesis, we briefly include the definition of OCDD. OCDD uses the class-attribute dependence information as the criterion for optimal discretization. Given a labeled dataset with *M'* training instances each of which has been preclassified into one of the K classes c_k ($k = 1, \ldots, K$), we could calculate the interdependence redundancy measure R between class label C and attribute A. Therefore, the discretization problem can be formulated as find the partition of attribute A such that the class-attribute interdependence redundancy measure $R(C : A)$ is maximized. Let Ψ represent the set of all possible finite partition schemes. Given class-attribute pair, one needs to find a $\psi_{\max} \in \Psi$ such that:

$$\forall \psi \in \Psi, R(C: A^{\psi_{\max}}) \geq R(C: A^{\psi})$$

OCDD adopts an iterative dynamic programming algorithm, which superlinearly converges to its optimal solutions, with the objective function to maximize $R(C : A) = I(C : A) / H(C : A)$.

We first employ OCDD to partition continuous attribute values into a finite number of intervals. From the feature-class dependence argument, we regard the mode as the class attribute in each attribute group. To begin with OCDD, the mode should be a discrete attribute. In case the mode is a continuous attribute, it should be discretized first. In general, if the number of intervals is not decided, in view of no prior information, entropy maximization [Wong and Chiu 1987] is used for the discretization. Once all the modes are discretized, other attributes follow using OCDD.

For each attribute other than the modes, it is partitioned by OCDD multiple times — each time with a different attribute group while treating the mode of that group as the class label to drive the discretization. For each partitioning, the partition result is associated with a degree of membership to an attribute group. After all continuous attribute values are discretized into a finite number of intervals, we can consider that the mixed-mode dataset contains only categorical data and the pattern discovery phase can be conducted.

5.4 Pattern Discovery

In this phase, pattern discovery [Wong and Wang 2003] method for categorical data could be applied readily. In an unsupervised manner, it detects high order patterns defined as statistically significant associations of 2 or more primary events from different attributes using the adjusted residuals d to test the significance of its occurrence against the independence assumption [Wong and Wang 2003]. The adjusted residue is a normalized statistical measure that

accounts for the deviation of the observed frequency of an association (order >2) from its expected default model of independence [Wong and Wang 2003]. If the association pattern is conditioned by the class attribute, it can be used as classification rule [Wang and Wong 2003]. The weight of evidence in information theory [Wang and Wong 2003] is used to quantify the evidence of the joined significant association rules to support or against a certain class membership. The definition of adjusted residue and weight of evidence measures have been given in section 4.2.2 in detail.

5.5 Experiments and Results

To verify the premises of how realistic the proposed approach tackles the discovery of patterns when applying to various types of mixed-mode data, appropriate experiments should be designed. In this section, we attempt to design a set of experiments with selected datasets of various types to test our premises.

First, we will design two comprehensive synthetic experiments with 2 sets of stochastically data generated to test each of the premises proposed. Experimental results are analyzed and compared to see whether our pattern discovery is consistent to the patterns we artificially implanted into the synthetic data.

Second, we will apply our pattern discovery method to various sets of UCI machine learning archive data [Asuncion and Newman 2007] to test the premises. Most selected datasets for these experiments are quite familiar to the data mining community. Since our method is unsupervised, the class labels contained in the selected datasets will be removed but regarded as the ground truth, though not absolute, for the examination of the performance of the proposed method. In this way, we could observe whether our method is able to

perform the pattern discovery tasks as anticipated and returns reasonable results even though the class labels of the datasets are excluded in our analysis.

Third, the proposed method is applied to a colon cancer gene expression dataset with its known class label removed. To calibrate the effectiveness of our method, after fuzzy attribute clustering and data discretization that optimizes the intra-group interdependence, we bring back the class labels to the gene expression dataset and assess the strength of the association patterns discovered through the classification performance using the patterns/rules discovered from the discretized events.

Fourth, two sets of large real world data of mixed-mode nature are analyzed through the proposed approach. The first set is a meteorological data collected from 6 stations located over a wide area for a relatively long period of time. The second set is taken from an operational database related to the processing of coke and gasoline from a delay coking plant. This mixed-mode dataset contains data gathered from site sensors, regulators and controllers. These 2 datasets were collected and provided by Sinocan Intellitech Ltd [Wu 2010] with the help of domain experts. Though these 2 sets of data are complex and not containing class labels, they are backed by adequate domain knowledge for affirmation of the analytical results to see whether the subtle operational patterns could be discovered by the proposed approach which does not require any prior knowledge.

These designed experiments are going to answer the following questions.

1) Is it possible to optimally cluster a large mixed-mode database containing attribute of categorical and continuous values?

2) Within a correlated dataset, does the existence of certain attributes (modes) reflect the characteristic of attribute groups (clusters) and these certain attributes act like class labels?
3) Within an overlapping dataset, does the existence of certain attributes have strong correlation to more than one attribute group, or associate with patterns which might overlap different attribute groups?
4) Could the proposed method be able to obtain such groups and attributes addressed in 1) attribute clustering, 2) mode identification and, 3) attribute cluster fuzzification?
5) If the attribute clustering, mode identification and attribute cluster fuzzification are operated, how effective is the discretization of the continuous data driven by such information (i.e. optimizing the interdependence between the modes and the continuous attributes)?
6) Once the mixed-mode dataset is transformed into one containing only discrete valued events, how effective is the pattern discovery and data mining methods when applying to it?

In answering these questions through the application of our proposed approach in the experiments, we hope new light could be shed to those difficult and not yet properly solved problems.

5.5.1 Experiments on Synthetic Datasets

This set of experiments is designed to verify the applicability of the proposed approach to mixed-mode datasets. It attempts to answer questions 1) to 6). It tries to demonstrate the role of the representative attribute (mode) in

inducing discretization of the continuous data just like the class attribute would even when the class label is absent and also how significant patterns overlapping different clusters cannot be found in crisp clusters but could be found in overlapping or fuzzy clusters.

5.5.1.1 Synthetic Dataset I

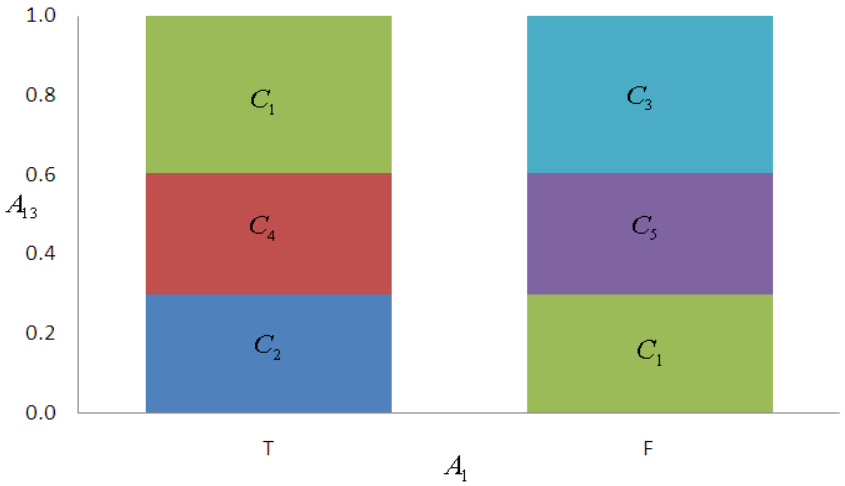


Figure 5.1. Imposition of intrinsic classes by adjusting the attribute values of certain attributes.

The synthetic data set I is composed of 20 attributes in which 5 of them are discrete and 15 of them are continuous. 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 5.1. 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 $A_{13} < 0.5$; "*F*", otherwise.
- A_3: "*X*" if $A_{13} < 0.5$; "*Y*" if $0.5 \leq A_{13} < 0.75$; "*Z*", otherwise.
- A_4: "*X*" if $A_6 < 0.3$; "*Y*" if $0.3 \leq A_6 < 0.6$; "*Z*", otherwise.
- A_5: "*Y*" if $A_6 < 0.3$; "*Z*" if $0.3 \leq A_6 < 0.6$; "*X*", otherwise.
- A_6-A_7: uniformly distributed within [0, 0.5] if A_1 = "*T*"; uniformly distributed within (0.5, 1], otherwise.
- A_8-A_{12}: uniformly distributed within [0, 0.5] if A_1 = "*F*"; uniformly distributed within (0.5, 1], otherwise.
- A_{14}-A_{17}: uniformly distributed within [0, 0.3) if $A_{13} < 0.3$; uniformly distributed within [0.3, 0.6) if $0.3 \leq A_{13} < 0.6$; uniformly distributed within [0.6, 1], otherwise.
- A_{18}-A_{20}: uniformly distributed within [0.3, 0.6) if $A_{13} < 0.3$; uniformly distributed within [0.6, 1] if $0.3 \leq A_{13} < 0.6$; uniformly distributed within [0, 0.3), otherwise.

Using this scheme to generate the synthetic data set, it is clear that A_1 and A_{13} are two representative attributes (modes) correlating with the attribute groups (clusters) of $\{A_4\text{-}A_{12}\}$ and $\{A_2, A_3, A_{14}\text{-}A_{20}\}$ respectively. Regardless of the class membership of each tuple, if such correlation can be revealed, one should seek the most representative attribute of each attribute group to drive the discretization of the continuous attributes. In our experiment, we generated 250 tuples where each class contains 50 tuples in the synthetic data set. Noises are then added by replacing 25 percent of the tuples with randomly generated values within the range of the corresponding attributes.

Firstly, the interdependence redundancy measure R as defined in definition 3-1 between each pair of discrete attributes, each pair of continuous attributes and each pair of discrete and continuous attributes is calculated.

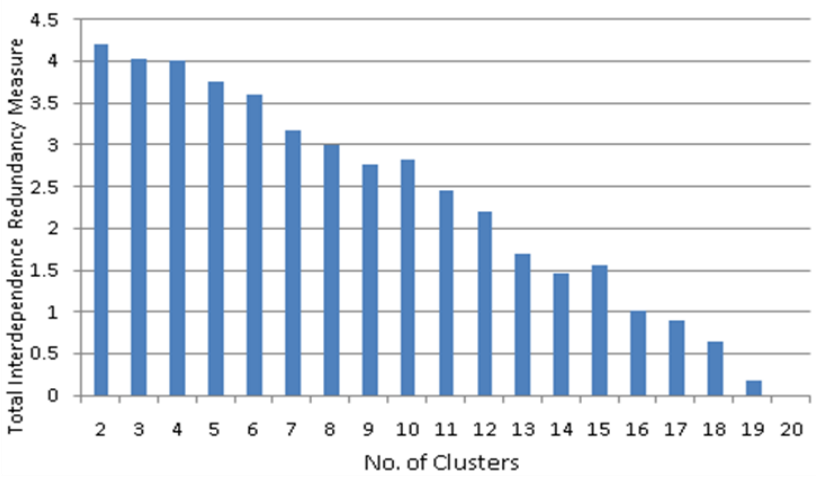


Figure 5.2. The total interdependence redundancy measure across the clusters found in synthetic dataset I.

As shown in Figure 5.2, the optimal attribute cluster configuration (no. of attribute clusters) obtained by MACA is two (k =2). MACA identifies two attribute clusters: {A_1, A_4, …, A_{12}} and {A_2, A_3, 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 MACA to reveal the correlation between the mixed-mode attributes embedded in the synthetic dataset. It was found that A_1 is the mode of the first attribute cluster whereas A_{13} is the mode of the second attribute cluster. It indicates that the attributes with the most intrinsic governing or classificatory characteristics as reflected via their statistical inter-dependence with other attributes in their group are found as the modes.

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 a C5.0 decision tree algorithm. 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. In comparison, the synthetic data set was also discretized by OCDD making use of the class label information. OCDD is experimentally proven to be a very effective discretizer when comparing with other unsupervised discretization algorithms like Equal Width, Equal Frequency and Maximum Entropy [Wong and Chiu 1987] and supervised discretization like CADD [Ching, Wong and Chan 1995]. The classification accuracy of C5.0, which is an existing class dependent algorithm, on data discretized by OCDD and that of our proposed method, which does not require any a priori knowledge, on the same data is 74% and 83.67% respectively. The comparison results show that the proposed method surprisingly reached higher 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 better performance (13 leaf nodes and 10 non leaf nodes) while C5.0 produced significantly more nodes (17 leaf nodes and 10 non leaf nodes) when using the discretization scheme of OCDD which makes use of class label.

5.5.1.2 Synthetic Dataset II

This experiment is designed to calibrate the proposed approach in verifying the premise that some significant patterns may overlap different crisp clusters and the proposed approach allows overlapping relationship to be found among attribute groups. Thus, patterns which may overlap crisp attribute clusters could be found. The synthetic data set II is composed of 20 attributes: 5 discrete

and 15 continuous (Figure 5.3). Let us denote the attributes as $A_1, \ldots, A_{20}$. 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"}. $A_6, \ldots, A_{20}$ are continuous attributes which can take on values in $\{0 \leq \Re \leq 1\}$ where $\Re$ is a real number.

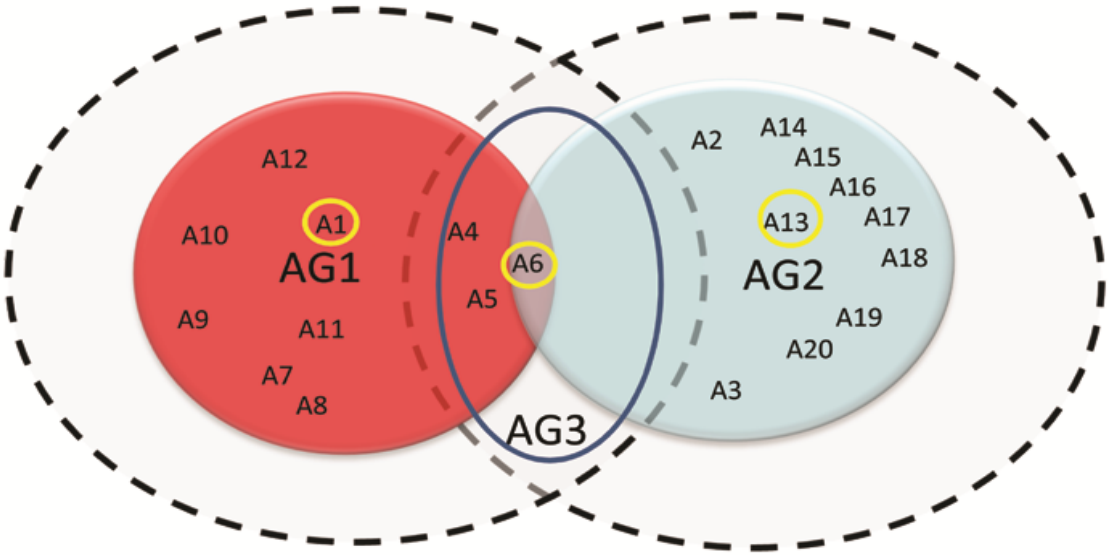


Figure 5.3. Attributes of the synthetic data II.

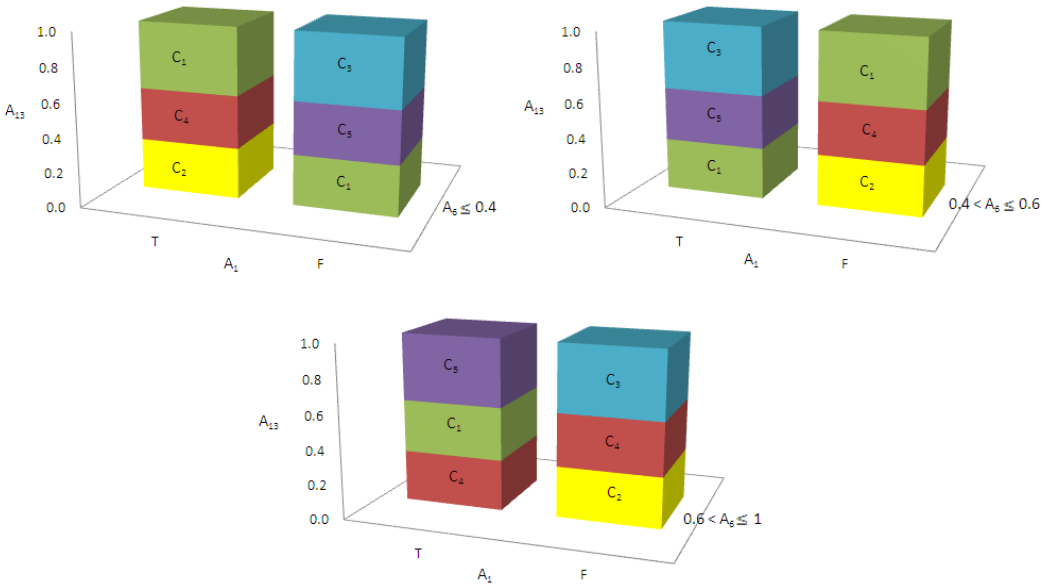


Figure 5.4. Imposition of intrinsic classes by adjusting the attribute values of certain attributes.

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, A_6 and A_{13} among the tuples (Figure 5.4). As in our designed experiment, attribute values A_1, A_6 and A_{13} of each tuple are able to determine the class membership. For overlapping attribute cluster relationship, $A_{4\text{-}6}$ are associated with both attribute cluster 1 and attribute cluster 3 with different degrees of membership. From Figure 5.3, we observe that A_6 is the mode of attribute cluster 3, AC_3, and $\mu_{AC_1}(A_6) > \mu_{AC_2}(A_6)$. A_1 and A_{13} is the mode of attribute cluster 1, AC_1, and attribute cluster 2, AC_2, respectively. For values of other attributes including A_2, …, A_{12} and A_{14}, …, A_{20}, they are generated in the following manner.

- A_2: "T" if $A_{13} < 0.2$; "F", otherwise.
- A_3: "X" if $A_{13} < 0.2$; "Y" if $0.2 \leq A_{13} < 0.4$; "Z", otherwise.
- A_4: "X" if $A_6 < 0.3$; "Y" if $0.3 \leq A_6 < 0.6$; "Z", otherwise.
- A_5: "Y" if $A_6 < 0.2$; "Z" if $0.2 \leq A_6 < 0.4$; "X", otherwise.
- A_6: uniformly distributed within [0, 0.7] if A_1 = "T" and $A_{13} < 0.5$; uniformly distributed within (0.3, 0.8] if A_1 = "T" and $A_{13} >= 0.5$; uniformly distributed within [0, 1], otherwise.
- A_7: uniformly distributed within [0, 0.5] if A_1 = "T"; uniformly distributed within (0.5, 1], otherwise.
- $A_{8\text{-}12}$: uniformly distributed within [0, 0.5] if A_1 = "F"; uniformly distributed within (0.5, 1], otherwise.
- $A_{14\text{-}17}$: uniformly distributed within [0, 0.3) if $A_{13} < 0.3$; uniformly distributed within [0.3, 0.6) if $0.3 \leq A_{13} < 0.6$; uniformly distributed within [0.6, 1], otherwise.

- $A_{18\text{-}20}$: uniformly distributed within [0.3, 0.6) if $A_{13} < 0.3$; uniformly distributed within [0.6, 1] if $0.3 \leq A_{13} < 0.6$; uniformly distributed within [0, 0.3), otherwise.

In our experiment, 1800 tuples of mixed mode attributes are generated. C_1, C_2, C_3, C_4 and C_5 contain 500, 300, 300, 400 and 300 tuples respectively. For practicality, 25% noise is added to the data by replacing 450 tuples with random values. We first use Mixed Mode ACA (MACA) to obtain attribute clusters, modes and optimal cluster configuration (k) in order to reveal the correlated relationship. Then we use Fuzzy Mixed Mode ACA (FMACA), with fuzzification parameter m = 1.5, to obtain the degree of membership of each attribute A_i to each attribute cluster, AC_j, to reveal the overlapping relationship.

Table 5.1. Attribute clusters discovered by MACA

AC	M	SR	Item
1	A1	1.7159	A1, A8, A7, A11, A12, A10, A9
2	A13	1.0494	A13, A2, A3, A16, A15, A17, A14, A18, A20, A19
3	A6	0.5978	A6, A4, A5

Key: AC – Attribute Cluster. M – Mode / Representative Attribute. SR - Sum of the Multiple Interdependence Measure.

Table 5.2. Degree of membership of attribute in attribute cluster

Attribute	AC_1	AC_2	AC_3
$\boldsymbol{A_4}$	8.9602%	0.0005%	91.0393%
$\boldsymbol{A_5}$	2.4429%	0.0031%	97.5540%
$\boldsymbol{*A_6}$	0.3961%	0.0001%	99.6037%

Key: $*A_6$ is the mode of this attribute cluster. *AC* – Attribute Cluster.

As shown in Table 5.1, MACA reveals the attribute grouping without prior knowledge (with class label excluded). It is worth to note that without fuzzification, however, it cannot show how some attributes are related among different attribute clusters since an attribute is a member of only one cluster. By FMACA, it shows that 3 attributes are indeed overlapping with AC_1, AC_2 and AC_3 with different degrees of membership as shown in Table 5.2.

5.5.2 Experiments on UCI Machine Learning Archive Datasets

5.5.2.1 Iris Plants

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. It attempts to answer questions 2), 4), 5) and 6). Because of the transparency characteristics of pattern discovery, new light could be shed to reveal how the representative attributes are related to the correlated aspects of the attributes and also with the class labels. The Iris data set [Asuncion and Newman 2007] with 150 samples and 4 numeric attributes contains 3 classes (Setosa, Versicolour and Virginica) of 50 instances each, where each class refers to a type of iris plant. The 4 numeric attributes are sepal length, sepal width, petal length and petal width.

We first use the class attribute to discretize the rest of the attributes and obtain the classification rate by discover*e, a commercial tool for pattern discovery [Wang and Wong 2010]. The classification rate for the class labels from the data set with labels retained is 96%.

We then remove the class labels from the data set and assume that each of the remaining four as the class attributes (representative attributes) in turn to drive the discretization of all the continuous data and conduct the classification afterward. The classification rate obtained by considering sepal length, sepal width, petal length and petal width as the governing ones is 76.67%, 64.67%, 92% and 92% respectively.

From the classificatory results obtained, it is clear that the last two attributes, petal length and petal width, could be considered as the representative attributes as they both yield the highest classification rate even without the class labels. To reveal how the representative attribute relates to the correlated aspects of the other attributes, we discretize the four attribute a) driven by the class label and b) driven by the representative attribute, which is the last attribute, when the class label is taken from the data set. To our surprise the discretization results driven by the last attribute is identical to those driven by the class labels.

After converting all the data into discrete valued events, pattern discovery methods can then be applied. Some examples of patterns discovered after the Iris data is discretized include a) if sepal width is within [1, 3] and petal length is within [0.1, 3], then it is classified as Setosa, b) if sepal width is within [3, 4.9] and petal length is within [0.1, 1], then it is classified as Versicolour and c) if petal width is within [6.3, 7.9] and sepal width is within [4.9, 6.9], then it is classified as Virginica.

5.5.2.2 Mushroom

The mushroom data is a dataset with categorical data only. It is composed of 8,214 samples with 23 attributes. This dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family. 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. Thus, the dataset contains 2 classes (edibility *e* and poisonous *p*). Containing only 2 classes, which is relatively small, this dataset is used to explore the possibility of the existence of attribute subgroups, each of which may govern a certain aspect of the characteristics of

the mushrooms. We attempt to use this experiment to answer questions related to 1), 2), 4) and 6). In particular, the objectives of this experiment are: a) to explore the ranking of the attributes according to their normalized significant interdependent redundancy, normalized *SR*, in the dataset with class label included; b) to compare the ranking of the attributes in the dataset with class label excluded with the ranking listed in (a); c) to compare the attributes with highest normalized *SR* with the class attributes; d) to show that in a normal setting the attribute with highest normalized *SR* value is also the attribute that renders 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 proposed 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.

Table 5.3 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 9^{th} based on the normalized *SR*. This implies that some of the attributes chosen are not necessarily closely related to the class attribute proposed by the biologists.

Table 5.3. Attributes from mushroom data (with class label included) ranked according to normalized *SR*. Note that the class label is not ranked top.

Ranking	Attributes	*R*	Normalized *SR*
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 5.4 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 5.4 shows the ranking of the attributes according to normalized *SR* 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 5.4 is ranked fourth in Table 5.3. The top eight ones in Table 5.4 remain the same as those in Table 5.3 indicating the consistence of the governing attributes in relation with the class label attribute.

A series of experimental runs were conducted next treating each of the attribute as the governing one in turn and obtain the classification rate (CR) accordingly. The attributes are then ranked according to the classification rates and the ranking results were compared with those ranked according to the normalized *SR* values obtained for the attributes in that group (Table 5.5).

In Table 5.5, we observe from the normalized *SR* ranking that the two attributes, the ring-type and stalk-root, rank top all other attributes. They are ranked first and fourth in Table 5.3 when the class labels are present. That the ranking of the Class Attribute is not ranked top according to normalized *SR* indicates that its interdependence 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 governing more in the sense that they have higher interdependence with other attributes in the group.

Table 5.5. 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 *SR*
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

Classification experiments were then conducted on these two sets of data (one with class labels and the other without class labels). We first conducted supervised learning of the data according to the class labels given and as expected, obtained 100% rate of correct classification. Then, we moved 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 representative 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. We next took "stalk root" as the representative attribute and again obtain 100% classification rate. 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/representative 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 *SR* value, the following observations are made. As pointed in the reference source [Asuncion and Newman 2007], 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 MACA 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 first apply MACA to the dataset with class labels 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 5.6 gives the attribute groups discovered in the experiment where class labels are included. 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 normalized *SR* ranking of the rest of the three attributes in the group are not too high (cap-color ranked 13^{th}, gill size 15^{th} and ring-number 19^{th}). It shows that as far as the "edibility" and "poisonous" properties are concerned, 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 are no obvious class labels available. Unless we have full knowledge ahead of time, for a given 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.

Table 5.6. Attribute clusters of mushroom data with class label included.

Attributes	*R*	Normalized *SR*
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	*R*	Normalized *SR*
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	*R*	Normalized *SR*
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

Three cluster configurations are the optimal. They are tabulated with the attributes in each cluster ranked according to the normalized *SR* value of the attribute of the group.

Table 5.7 shows the results of the experiment of the dataset without class labels. It is worth to 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 representative attributes, we observe in Table 5.5 that although the normalized *SR* value for ring-type is a little higher (0.1357 > 0.1231), yet the distribution of the categories it encompassed is less even (uneven distribution vs. even distribution) when comparing the classification rate of their categories. Hence as far as the representative characteristic of these two attributes in the attribute groups is concerned, the latter seems to offer a better candidate. This will be explored by our future research.

Table 5.7. Attribute clusters of mushroom data with class label excluded.

Attributes	*R*	Normalized *SR*
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	*R*	Normalized *SR*
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

Two clusters is the optimal attribute cluster configuration. They are tabulated with the attributes in each cluster ranked according to the normalized value of the attribute of the group.

When we look closer at the attributes forming these two correlated groups as shown in Table 5.7, it is observed that all the attributes associated with the class label (Table 5.6) reside in the second group headed by the mode of stalk-root in Table 5.7. 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 our proposed attribute clustering algorithm, our proposed mode identification algorithm, our proposed mode driven discretization algorithm and classification procedure presented in this chapter.

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 as well as the identification of modes (or other top ranked attributes) in the original dataset and the clustered attribute groups render considerable insights into the inherent make-up 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.

5.5.2.3 Adult

This data set obtained by UCI Machine Learning Archive [Asuncion and Newman 2007] was extracted from US Census Bureau database. It contains 48,842 instances of a mix of continuous and discrete data with 14 attributes. It has been used for prediction task whether a person makes over 50K a year or not. This experiment is used: a) to demonstrate the existence of attribute subgroups in the mixed-mode data set; b) to illustrate the attainment of attribute cluster

configuration and the grouping of cluster items in situations with or without class label; c) to show the classification characteristics of various attributes in different attribute groups found by MACA; and d) to show that the attribute with highest normalized MR, 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. It attempts to answer questions 1), 2), 4), 5) and 6). The experiment results do show that the mode in each attribute group/cluster can be considered as the most representative attribute to drive the discretization of continuous attributes in the attribute group/cluster.

In order to demonstrate the effectiveness of the proposed method in extracting the same intrinsic information inherent in the classes, we experimented on the dataset with class label excluded and those with class label included. Based on S*R* values, ACA found the optimal cluster configurations 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 our proposed method, no class information is required; nevertheless, the results reported in Table 5.8 show that even without class information, our proposed method and ACA are able to group interdependent attributes together.

Table 5.8. The attribute clusters and their corresponding modes obtained by ACA

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
Key: *- The mode of the attribute group. A mode is with the highest MR 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, the *MR* values and the classification performance of their attributes are tabulated in Table 5.9.

Table 5.9. Attribute clusters of with class label excluded

AG	A	T	MR	CA (%)
1	* native-country	D	0.0952	89.59
	Race	C	0.0898	84.43
	Fnlwgt	C	0.0083	5.41
2	* education	D	0.8263	71.09
	Workclass	D	0.8218	57.69
	Occupation	D	0.2051	20.94
	education-num	C	0.1173	-
3	* relationship	D	0.6251	72
	# marital-status	D	0.5525	74.78
	Sex	D	0.2465	68.95
	Age	C	0.2229	-
	^ capital-gain	C	0.1100	99.51
	^ capital-loss	C	0.0495	95.33
	hours-per-week	C	0.0313	14.54

Key: *- The attribute marked with "*" is the mode of 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. AG- Attribute Group. A- Attribute. T- Type. MR- Multiple Interdependency Redundancy Measure. CA- Classification Accuracy. D- Discrete. C- Continuous.

Once the mixed-mode Adult data set is transformed into one containing only categorical events using the proposed method, pattern discovery methodology could be readily applied to the transformed data set. Some examples of patterns discovered after the Adult data is discretized include a) if education is "HS-grad" and education-num is within [9, 10], then income is "<=50K" b) if marital-status is "Married-civ-spouse" and relationship is

"Husband", then income is ">50K" and c) if marital-status is "Married-civ-spouse" and hours-per-week is within [40, 99], then income is ">50K".

5.5.3 Experiment on Colon Cancer Gene Expression Dataset

The colon-cancer gene expression dataset [Alon et al. 1999] is chosen for analysis due to its public availability. In this experiment, we attempt to use it to answer questions 1) to 6). The dataset is composed of 62 samples and 2,000 genes and is represented by a 62 tuples x 2000 gene expression table. Each sample (tuple) is pre-classified into either normal or cancerous.

Since our method is unsupervised, we remove the tissue class label in the initial experimental phase. We first cluster the genes to obtain the gene clusters. As our FMACA supports mixed mode data type, it is unnecessary to discretize the continuous data in the first place. As expected, FMACA found 7 optimal gene clusters as same as the result reported by [Au et al. 2005]. The experimental result shows that our pattern discovery methodology is able to uncover the correlated genes (attributes) and patterns without making use of class information. The top 5 genes of each cluster as shown in Table 5.10 are selected for classification in the second experimental phase.

Table 5.10. Top 5 genes in each of the 7 clusters found in the colon-cancer dataset [Au et al. 2005]

Accession Number	Name
H05814	PUTATIVE ATP-DEPENDENT RNA HELICASE C06E1.10 IN CHROMOSOME III (Caenorhabditis elegans)
X02874	Human mRNA for (2'-5') oligo A synthetase E (1,6 kb RNA)
U33429	human K+ channel beta 2 subunit mRNA, complete cds
H22579	INTEGRIN ALPHA-6 PRECURSOR (Homo sapiens)
H25940	PUTATIVE SERINE/THREONINE-PROTEIN KINASE PSK-H1 (Homo sapiens)
T73092	EUKARYOTIC INITIATION FACTOR 4A-I (Homo sapiens)
R26146	NUCLEAR FACTOR NF-KAPPA-B P105 SUBUNIT (HUMAN)
T90851	ADP-RIBOSYLATION FACTOR-LIKE PROTEIN 4 (Rattus norvegicus)
R93337	HOMEOTIC GENE REGULATOR (Drosophila melanogaster)
T69446	EUKARYOTIC INITIATION FACTOR 4A-I (HUMAN)
M26383	Human monocyte-derived neutrophil-activating protein (MONAP) mRNA, complete cds
U34252	Human r-aminobutyraldehyde dehydrogenase mRNA, complete cds
T59162	SELENIUM-BINDING PROTEIN (Mus musculus)
M27749	IMMUNOGLOBULIN-RELATED 14.1 PROTEIN PRECURSOR (HUMAN)
T54341	P25886 60S RIBOSOMAL PROTEIN L29
T51849	TYROSINE-PROTEIN KINASE RECEPTOR ELK PRECURSOR (Rattus norvegicus)
D13243	Human pyruvate kinase-L gene, exon 12
X52008	H.sapiens alpha-2 strychnine binding subunit of inhibitory glycine receptor mRNA
R48936	GLYCOPROTEIN VP7 (Chicken rotavirus a)
X14968	Human testis mRNA for the RII-alpha subunit of cAMP dependent protein kinase
T90036	CLASS I HISTOCOMPATIBILITY ANTIGEN, E-1 ALPHA CHAIN PRECURSOR (Pongo pygmaeus)
R81170	TRANSLATIONALLY CONTROLLED TUMOR PROTEIN (Homo sapiens)
X67235	H.sapiens mRNA for proline rich homeobox (Prh) protein
L20469	Human truncated dopamine D3 receptor mRNA, complete cds
T63133	THYMOSIN BETA-10 (HUMAN)
T92451	TROPOMYOSIN, FIBROBLAST AND EPITHELIAL MUSCLE-TYPE (HUMAN)
H11460	GOLIATH PROTEIN (Drosophila melanogaster)
H23975	IG ALPHA-1 CHAIN C REGION (Gorilla gorilla gorilla)
R70030	IG MU CHAIN C REGION (HUMAN)
D10522	Human mRNA for 80K-L protein, complete cds. (HUMAN);contains element TAR1 repetitive element
H71627	VITELLOGENIN A2 PRECURSOR (Xenopus laevis)
X74795	H.sapiens P1-Cdc46 mRNA
T55840	TUMOR-ASSOCIATED ANTIGEN L6 (Homo sapiens)
D17400	Human mRNA for 6-pyruvoyl-tetrahydropterin synthase, complete cds
R71585	EBNA-2 NUCLEAR PROTEIN (Epstein-barr virus)

In the second experimental phase, we first discretize the continuous gene expression levels by the proposed discretization strategy and then put back the tissue class label as an attribute to the discretized dataset. Since modes of all attribute clusters are continuous, we need to first discretize them first and then use them to drive the discretization of other attributes. To discretize continuous modes unsupervisedly, we employ entropy maximization algorithm. Due to the relatively small sample size, we consider partitioning the modes into 3 states: highly expressed (H), normally expressed (N) or lowly expressed (L). This set of preprocessed data is trained by popular classification methods for building classifiers. We compare our classification results with those reported in [Au et al. 2005].

The classification accuracy of C5.0 and our pattern discovery using data preprocessed by our proposed method is 85.48% and 91.94% respectively while those using the data preprocessed by ACA as reported by [Au et al. 2005] is 91.9% and 100% respectively. From the results, it shows that the proposed method is comparable to that requiring prior class information whereas ours does not. The significance of this experiment is that, even without using class labels, the intrinsic interdependence gene expression levels are brought out 1) to reveal the inherent relationship of gene groups, 2) to select the most representative genes in each group, 3) to use their combined relationship to relate back to the class relation and achieve a high rate of classification and, 4) to use a fuzzy membership function to weigh the overlapping attributes for optimal discretization. As a consequence, the discretized data returned by the proposed method obtaining high classification rate after putting back the class labels is a

realization of the meaningfulness and effectiveness of the proposed method. The top 5 patterns and rules are shown in Table 5.11 for reference.

Table 5.11. Top 5 patterns and rules discovered in colon-cancer gene expression dataset

Pattern	Adjusted Residual
T69446 = H, T73092 = H, Cancer	6.43
U34252 = H, T92451 = H, Normal	5.87
M27749 = L, M26383 = L, T92451 = H, Normal	5.39
M27749 = L, T63133 = L, T92451 = H, Normal	5.38
H22579 = H, H05814 = H, Cancer	5.36

Rule Condition	Result	WOE
X02874 = H and U33429 = H and M26383 = H	Cancer	6.43
X02874 = H and U33429 = H and T92451 = L	Cancer	5.87
X02874 = H and U33429 = H and X74795 = H	Cancer	5.39
X02874 = H and H25940 = H and T63133 = H	Cancer	5.38
X02874 = H and R26146 = H and T59162 = L	Cancer	5.36

Key: WOE – Weight of evidence.

5.5.4 Experiments on Real World Datasets

After examining the performance of the proposed method, it is applied to 2 sets of large real world data of mixed-mode nature. The data was collected by the Sinocan Intellitech Ltd [Wu 2010] 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.

5.5.4.1 Meteorological Database

The meteorological (MET) database is a large database consisting of 8,784 samples and 43 attributes of which 18 are categorical and 25 are continuous. The MET data was taken from 5 different surface stations over a one-year-long period (8760 records) in the great urban region of Guangzhou City, Guangdong province, China. The types of the meteorological parameters (attributes) collected from each surface station include 6 discrete attributes and 5 continuous attributes. 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. The five surface stations denoted by the alphabets S = {A, B, C, D, E} are stations as listed in Figure 5.5. Station A, B, C, D and E is Guangzhou metropolis, Foshan city, Shenzhen city, Dongguan city and Zhongshan city respectively. The description of data collected by each station is listed in Table 5.12.

We applied the proposed approach on this set of meteorological data. The sum of significant multiple redundancy of the clustering process for various attribute cluster configurations is plotted on figure 5.6. It is obvious that a local

optimal attribute cluster configuration would consist of 5 clusters of MET parameters.

Figure 5.5. Guangzhou urban region (GGA).

Table 5.12. Attribute description of the MET database.

Attribute	Name	Types	Notes
MM	Month	Discrete	Month
DD	Day	Discrete	Day
HH	Hour	Discrete	Hour
***S*1**	TC	Discrete	Total Cloudiness
***S*2**	LC	Discrete	Lower Cloudiness
***S*3**	DBT	Continuous	Dry Bulb Temperature
***S*4**	DPT	Continuous	Dew Point Temperature
***S*5**	RH	Continuous	Relative Humidity
***S*6**	SP	Continuous	Site Pressure
***S*7**	WD	Discrete	Wind Direction
***S*8**	WS	Continuous	Wind Speed
where *S* = {A, B, C, D, E} corresponds to a set of 5 surface stations.			

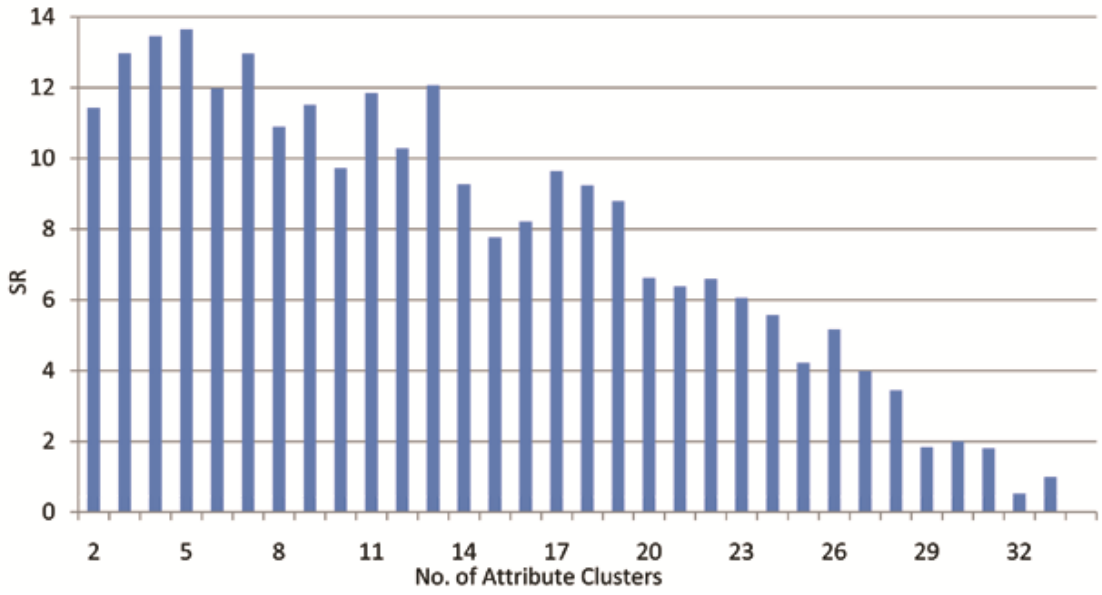


Figure 5.6. The plot of the sum of significant MR of MET.

Table 5.13 displays that after filtering 9 redundant attributes, the mixed-mode meteorological database with 34 attributes has been clustered into 5 sub-groups. These clusters have been labeled by domain experts. 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 5.13. Attributes in the attribute clusters of the optimal cluster configuration.

AG	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)
Key: *- The attribute marked with "*" is the mode of the attribute group. AG- Attribute Group. C- Continuous Attribute Group. D-Discrete Attribute Group. M- Mixed-Mode Attribute Group	

From the patterns discovered by our method, significant features within the data collected from the surface stations have been found, complying 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 5 stations. Regarding the last cluster group, all of the attributes therein reflect local characteristics which are significantly influenced by the local geographical feature such as land use and land coverage.

After the investigation of the correlated behaviors of the 5 stations, we look into the overlapping relationship among them. Table 5.14 reports the fuzzy degree of membership of each attribute to each attribute cluster. The first 4 of 5 groups have very strong modes with the highest degrees of member (rank top in each group) to their own groups. This implies that even though these clusters are overlapped, each mode is intrinsically representing a subspace (group). All

modes have small values of degrees of memberships to other attribute clusters, indicating that the modes discovered are well separated in the entire data space. It is not difficult to observe that some attributes are actually dependent on multiple attribute groups. Figure 5.7 visualizes the overlapping effects of the result.

Table 5.14. Fuzzy degree of membership of each attribute to each attribute cluster of MET

AG	A	1) Relative Humidity	2) Wind Direction	3) Total Cloudiness	4) Site Pressure	5) Dry Bulb Temperature & Wind Speed
AG1	*B5	99.8025%	0.0428%	0.0986%	0.0539%	0.0022%
	C5	98.7661%	0.2262%	0.5955%	0.3996%	0.0127%
	D5	98.5978%	0.3334%	0.6847%	0.3693%	0.0148%
	E5	98.4846%	0.5831%	0.4948%	0.4054%	0.0321%
	A5	97.9425%	1.2015%	0.4999%	0.3219%	0.0343%
AG2	*C7	0.0428%	99.8375%	0.0189%	0.0997%	0.0011%
	B7	2.8136%	90.4941%	1.2994%	**5.3328%**	0.0601%
	E7	2.9959%	85.5852%	1.3667%	**9.8518%**	0.2004%
	D7	4.1159%	83.8774%	0.2116%	**11.7334%**	0.0617%
	A7	**8.7221%**	75.9601%	0.8284%	**14.2256%**	0.2638%
AG3	*C1	0.0985%	0.0189%	99.6229%	0.0654%	0.1944%
	B1	0.6279%	0.8304%	91.9044%	2.0611%	4.5762%
	D1	1.4897%	0.5674%	91.2329%	1.7118%	**4.9981%**
	A1	1.0673%	0.8718%	90.7600%	1.9930%	**5.3080%**
	E1	6.6258%	0.7811%	81.2527%	2.6565%	**8.6839%**
AG4	*A6	0.0531%	0.0982%	0.0646%	98.3570%	1.4271%
	D6	0.1027%	0.3227%	0.1503%	96.2184%	3.2059%
	E6	0.0938%	0.3366%	0.1511%	96.1615%	3.2570%
	B6	0.1589%	0.2501%	0.1647%	95.7533%	3.6730%
	MM	1.8872%	**6.2018%**	**7.7794%**	59.1474%	**24.9843%**
AG5	C8	0.0001%	0.0000%	0.0066%	0.0542%	99.9390%
	B8	0.0001%	0.0000%	0.0115%	0.0850%	99.9033%
	D8	0.0001%	0.0000%	0.0103%	0.1141%	99.8755%
	A8	0.0001%	0.0000%	0.0104%	0.1169%	99.8725%
	E8	0.0001%	0.0000%	0.0098%	0.1803%	99.8097%
	E3	0.0022%	0.0011%	0.2037%	1.3461%	98.4469%
	C3	0.0022%	0.0011%	0.2281%	1.3738%	98.3948%
	B3	0.0022%	0.0011%	0.2239%	1.3884%	98.3844%
	*A3	0.0022%	0.0011%	0.1920%	1.4274%	98.3773%
	D3	0.0022%	0.0011%	0.2304%	1.4048%	98.3616%
	HH	**18.1598%**	0.2721%	0.4443%	1.3180%	79.8057%
	DD	0.6152%	0.3747%	**9.4124%**	**13.4738%**	76.1240%
	C6	0.1971%	0.4737%	4.5769%	**32.9393%**	61.8129%
	A4	-1.#IND00%	-1.#IND00%	-1.#IND00%	-1.#IND00%	-1.#IND00%

Key: A – Attribute; AG – Attribute Group; The attribute marked with "*" represents the mode of the attribute group; The cell highlighted in gray color indicates relatively high degree of membership to the other group.

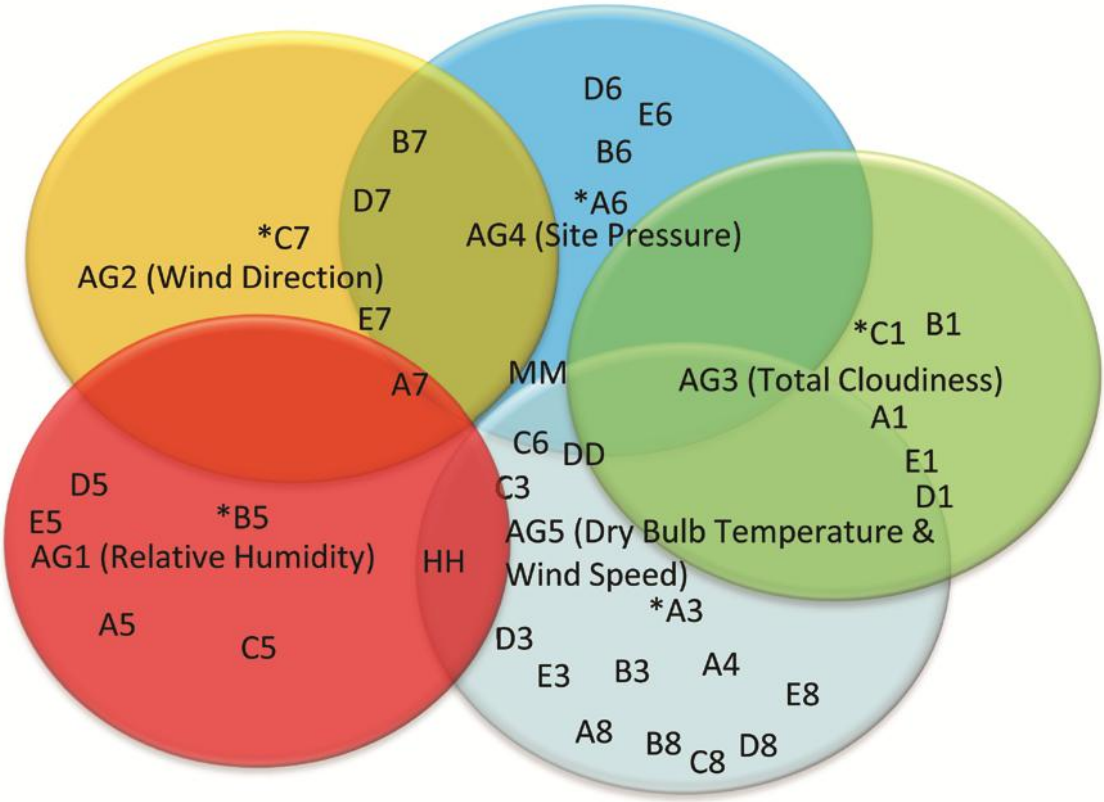


Figure 5.7. Semantic diagram of fuzzy / overlapping attribute clusters of MET.

From Figure 5.7, it is not difficult to discover the overlapping relationship of some attributes to several attribute groups. For instance, “A7” sensor for detecting wind direction in Guangzhou station is related to also “A6” - site pressure of Guangzhou and “B5” - relative humidity of Foshan. Geographically, these 3 sensors are located in close-by areas. Climatically, the difference in air pressure results in wind and wind effects mixing of the air mass. Strong winds promote better mixing and can bring either drier air or more moist air down to the surface. Therefore, the air pressure, the wind direction and the resulting moisture content of the free air are related to the surface relative humidity.

The discovered modes in these attribute clusters cover only 3 stations, namely A, B and C. This finding indicates that the remaining 2 stations, namely D and E, are in very weak position for the weather condition analysis.

5.5.4.2 Delay Coking Database

The dataset is taken from the delay coking unit (DCU) of the Sinopec SJZ Petro-Chemical refinery for about 5-month-long period. 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. 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. It is a semi-continuous thermal cracking process in which a heavy hydrocarbon feedstock is converted to lighter and more valuable products and coke. Its mechanism of coking can be broken down to three distinct stages as shown in Figure 5.8. The feed undergoes partial vaporization and mild cracking as it passes through a specially designed coking furnace. Since this is a set of very complex data taken directly from the delay cooking plant, there is no specific class information labeling the samples and it is relatively a large database. Provided that we have a certain degree of partial domain knowledge concerning DCU, this set of data will be ideal to challenge the usefulness and effectiveness of the proposed approach.

We applied the proposed attribute clustering method to cluster the database into sub-database containing subgroups of attributes. Figure 5.9 shows the plot of the sum of the significant *MR* values for different attribute cluster configurations. It is found that $k = 5$ would render a local optimal attribute cluster configuration. We next proceeded to discretize the continuous data for each cluster based on the mode discovered. The result of each attribute group revealing subtle operations is included in Figure 5.8.

Based on the five clusters discovered 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. Figure 5.8 displays the 5 clusters associated with the 3 stages of DCU. The number of attributes and distribution of the largest group, i.e. attribute group 1, 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 attribute group 2 and 4 control the output distributions of the two internal units, fractionator and coke drum. They are very important groups for the local performances of the processing usually referred to as performance factor.

The last discovered group, attribute group 5, is exactly associated with the critical safety mechanism designed for this pressure-temperature-mixed processing facility. Its mode actually controls the temperature condition as a triggering factor to activate the emergency release response.

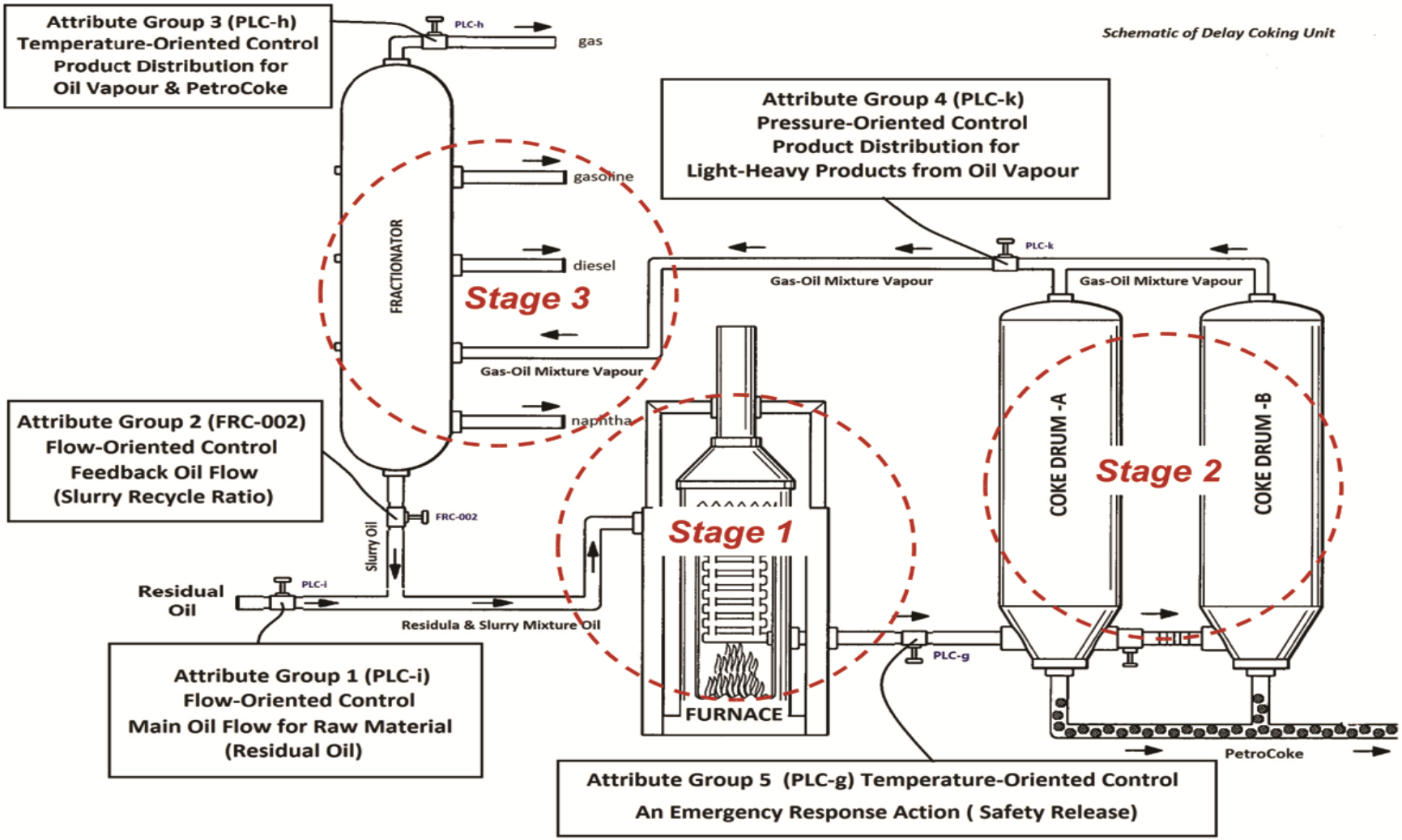


Figure 5.8. The schematic of delay coking unit.

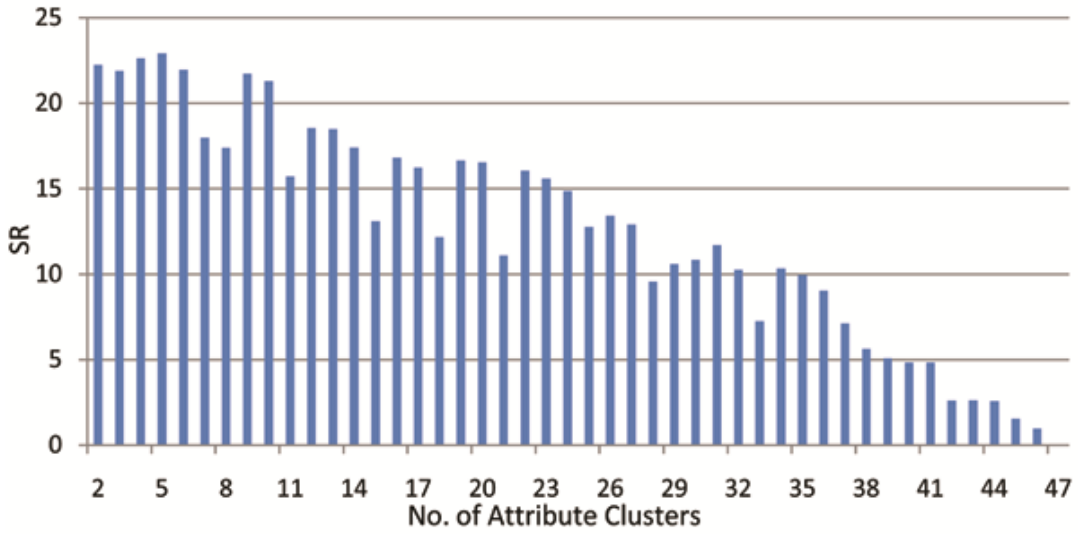


Figure 5.9. The plot of the sum of significant MR values against k, the number of attribute clusters.

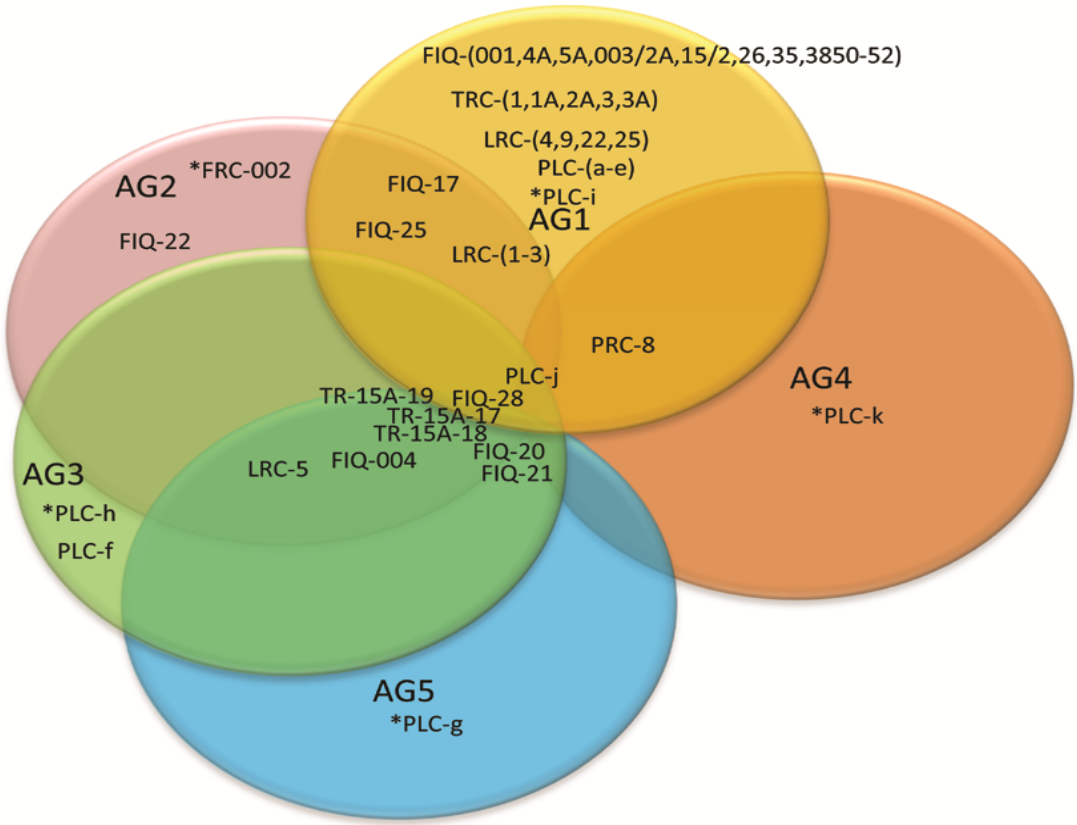


Figure 5.10. A schematic diagram of fuzzy / overlapping attribute clusters of DCU.

AG	A	1) Flow-Oriented Control of Main Oil Flow for Raw Materials (Residual Oil)	2) Flow-Oriented Control of Feedback Oil Flow Recycle Ratio	3) Temperature-Oriented Control of Production Distribution for Oil Vapor and Petro-Coke	4) Pressure-Oriented Control of Production Distribution for Light-Heavy Products from Oil Vapor	5) Temperature-Oriented Control of Emergency Response Action (Safety Release)
AG1	*PLC-i	99.8947%	0.0007%	0.0002%	0.1044%	0.0001%
	TRC-1A	99.8945%	0.0078%	0.0014%	0.0949%	0.0014%
	TRC-3A	99.8865%	0.0156%	0.0005%	0.0949%	0.0025%
	TRC-1	99.8694%	0.0189%	0.0116%	0.0949%	0.0053%
	TRC-3	99.8017%	0.0372%	0.0023%	0.1138%	0.0450%
	PLC-c	99.7512%	0.0062%	0.0006%	0.2417%	0.0004%
	PLC-e	99.7495%	0.0044%	0.0008%	0.2448%	0.0005%
	PLC-a	99.6404%	0.0064%	0.0016%	0.3500%	0.0016%
	PLC-b	99.5989%	0.0071%	0.0013%	0.3903%	0.0024%
	PLC-d	99.4736%	0.0091%	0.0029%	0.5113%	0.0031%
	LRC-22	99.4403%	0.4075%	0.0493%	0.0944%	0.0084%
	TRC-2A	99.3637%	0.4072%	0.0409%	0.0944%	0.0939%
	LRC-9	99.3429%	0.0551%	0.0052%	0.5910%	0.0059%
	FIQ-051	99.2323%	0.3842%	0.0302%	0.2715%	0.0819%
	LRC-4	98.5995%	0.5556%	0.0087%	0.8169%	0.0194%
	FIQ-26	98.4449%	1.2694%	0.0331%	0.2094%	0.0432%
	FIQ-052	98.1780%	0.0520%	0.0026%	1.7609%	0.0064%
	FIQ-003/2A	98.1373%	1.1886%	0.1676%	0.1411%	0.3655%
	FRC-4A	97.9217%	1.2879%	0.2959%	0.0930%	0.4016%
	FIQ-50	97.2628%	2.0896%	0.3252%	0.1221%	0.2003%
	FRC-001	96.9987%	2.4090%	0.3645%	0.1469%	0.0809%
	FRC-5A	96.8634%	2.6685%	0.1800%	0.1434%	0.1448%
	LRC-25	96.5333%	2.3797%	0.2523%	0.4191%	0.4156%
	FIQ-35	96.4823%	2.2596%	0.3458%	0.1366%	0.7756%
	FIQ-15/2	96.2054%	2.7066%	0.2918%	0.3933%	0.4029%
	FIQ-38	95.7667%	3.0716%	0.1303%	0.9433%	0.0881%
	LRC-1	88.2445%	8.6127%	1.0851%	0.7421%	1.3156%
	FIQ-17	84.6047%	7.9597%	2.5731%	0.1608%	4.7017%
	LRC-3	81.9288%	14.3761%	0.9613%	1.6029%	1.1310%
	FIQ-25	77.5091%	20.4298%	0.9062%	0.2536%	0.9012%
	LRC-2	66.8419%	26.7088%	1.9397%	3.1329%	1.3767%
	PLC-j	44.0822%	37.0202%	8.5554%	8.8515%	1.4907%
	FIQ-28	41.4797%	39.0487%	4.0648%	4.7824%	10.6244%
AG2	*FRC-002	0.0007%	98.1203%	1.1133%	0.0147%	0.7510%
	FIQ-22	0.0132%	93.4927%	1.6373%	0.0147%	4.8421%
	LRC-5	3.4462%	69.7859%	7.4612%	2.6614%	16.6454%
	FIQ-004	16.7241%	63.5683%	5.1786%	1.5412%	12.9879%
	FIQ-20	0.0002%	50.8589%	27.2105%	12.1276%	9.8028%
AG3	*PLC-h	0.0002%	1.0795%	95.1443%	0.0239%	3.7521%
	TR-15A-19	7.1582%	4.6154%	81.3287%	0.0479%	6.8497%
	PLC-f	0.0004%	2.2761%	77.5639%	0.0436%	20.1159%
	TR-15A-17	10.0596%	7.3063%	62.9045%	0.0781%	19.6514%
AG4	*PLC-k	0.1043%	0.0150%	0.0251%	99.8507%	0.0049%
	PRC-8	24.5699%	2.9804%	0.3730%	71.8727%	0.2040%
	FIQ-21	0.0001%	27.8705%	21.2283%	40.5431%	10.3581%
AG5	*PLC-g	0.0001%	0.7309%	3.7661%	0.0047%	95.4982%
	TR-15A-18	5.3593%	7.3183%	13.6367%	0.0188%	73.6670%

Key: A - Attribute; AG - Attribute Group; The attribute marked with "*" represents the mode of the attribute group.

Table 5.15. Fuzzy degree of membership of each attribute to each attribute cluster of DCU.

In investigating the overlapping relationship among the discovered 5 attribute clusters, we look at the results of attribute cluster fuzzification as shown in Figure 5.10 and Table 5.15. It is obvious that some attributes are related to multiple attribute clusters with significant high fuzzy degrees of memberships. Attribute group 1, which is the largest group, has the strongest overlapping relationship with the other 4 groups according to a high number of attributes having high degrees of memberships to other groups. This complies with the setting that attribute group 1 is a control unit for the entire processing system and has globally influenced almost all of the process parameters for the facility.

All of the 5 cluster groups with the patterns and mode attributes discovered provided us the strong pattern discovery and analysis evidence in revealing the underlying control principle in industrial systems.

5.6 Summary

In this chapter, we have introduced the proposed unsupervised pattern discovery approach for mixed-mode data that supports the discovery of useful data subspaces from the entire data space and how each individual attribute relates to each subspace. Its capability has been demonstrated by 9 sets of experiments with large datasets. It is first applied to 2 sets of synthetic data to verify its effectiveness in revealing correlated and overlapping behaviors of a database. Then applying it to 3 sets of popular data from UCI Machine Learning Archive demonstrates its effectiveness by uncovering the intrinsic class information inherent in their data attributes without relying on the given class information. To show its ability to obtain biological meaningful information from the gene expression data, a colon cancer microarray dataset is analyzed with positive results

reported. To challenge the proposed approach, 2 large sets of real data without class information are chosen to be analyzed. The discovered operational patterns and relationship are consistent with the domain knowledge acquired by domain experts. All these results confirm the pattern discovery algorithm has the ability to capture interesting and unknown information inherent in the mixed-mode database. The proposed approach can be regarded as a general framework for an operating platform that will not only help data management but also bring out the subtle knowledge trapped in the collected data in science, business and industry.

Chapter 6. Conclusions and Suggestions for Future Research

The research presented in this thesis was motivated by the real world challenges we are facing today. These challenges include but are not limited to: (1) an increasingly huge amount of raw mixed-mode type data from different areas which requires 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 KDD and decision support from overwhelming volume of discovered patterns; (3) the increasing demand of applications for discovering patterns in scientific, business and industry; and (4) limitations from most of the existing systems which are not general enough to solve problems on mixed-mode type 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 sequence, continuous, categorical and mixed-mode data. With the well defined problems and research objectives stated in Chapter 1, the developed proposed methods presented in Chapter 3, 4 as well as 5, and the broad applications on real world and industrial problems presented in Chapter 5, the contribution of the thesis research in theoretical and methodological perspectives as well as in real world applications have been conveyed. Through experimenting a series of computational experiments, the successful experimental results have supported the validity and the effectiveness of the proposed methods. The usefulness of the proposed methods in real world applications has been demonstrated by the intriguing and revealing

results obtained when applying to two large mixed-mode databases – the former one consists of a large set of meteorological data taken from a geographic area in Southern China and the later one is a set of massive multi-senor data taken from a delay coking plant.

6.1 Summary of Contributions

6.1.1 Theoretical Contributions

The theoretical contributions of the defined problems and the proposed methods could be summarized as follows.

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

   A theoretical framework has been developed for the discovery of high order patterns of sequence and mixed mode data at event level. By converting the sequence data into a relational table, this type of data can be treated under a general pattern discovery framework. In this unified pattern discovery framework, association patterns are defined as event associations which generalize the data mining method to cover special mixed mode type data. The experimental results of several synthetic data, famous UCI machine learning achieve data and two sets of real world industry data obtained show that under this theoretical framework, the discovered patterns could be organized, interpreted and easily understood. A merit of it is that the generated patterns/rules are presented in a form of high order patterns of event associations which cover various levels of event subspaces of lower dimensions.

2. Demonstrations of the necessity of fuzzy attribute clustering in large databases of mixed mode data and sequence data.

   From experimental results on extensive experiments on several data sets, the thesis provides supportive evidences that in large database, strong correlation among attributes do exist to form attribute clusters. This leads to our thought to make use of the information of how attributes/features of the data/samples are naturally associated to perform unsupervised data analysis. The revealed correlated group of attributes in the mixed mode database redefines the class-attribute relationship by introducing the concept of mode and thus enhances the discretization performance to optimize interdependence between modes and attributes in the group. The fuzzification concept introduced in additional to crisp attribute clustering furnishes the framework to detect patterns hidden in overlapping attribute clusters. This not only contributes a new problem to be stated but also create room for an algorithmic solution to tackle the task.

6.1.2 Methodological Contributions

1. A probabilistic approach to extract patterns in sequences without sequence alignment in an unsupervised manner.

   Based on the concept of sliding windows, an unlabelled sequence is divided into subsequences and then these subsequences are analyzed by an association discovery technique to detect statistical significant sequential patterns with gap allowance. The discovered patterns are summarized in a relational table for further analysis under the unified pattern discovery framework.

2. An algorithmic approach to compute interdependence redundancy measure (normalized mutual information) between mixed-mode attributes.

   In practice, one is always blocked by the obstacle to compute mutual information between continuous attributes due to the expensive computation to approximate the integration of two continuous attributes. In this thesis, an implementable algorithmic approach is defined for the computation of the normalized mutual information not only between a pair of discrete attributes but also between a pair of continuous attributes as well as between a discrete and a continuous attribute. Being tested over a number of data set from synthetic, UCI machine learning achieve and real world, the proposed measure is proven to be effective in finding correlated attribute clusters by *k*-mode ACA, the representative attribute (mode) of an attribute cluster and overlapping attribute clusters.

3. Modes discovery for attribute clustering and fuzzification.

   Using the interdependence redundancy measure, the mode can be obtained by searching for one with the highest sum of the measure with all other attributes in the same attribute cluster. The discovered mode holds the strongest interdependence to all other attributes in the group, can then be treated as a representative of the group and use it just like a class label to drive the discretization of other continuous attributes. The modes are also important to the fuzzy attribute clustering for the calculation of the degree of membership of an attribute to an attribute cluster.

4. Discretization of continuous attributes.

One major challenge in discretization is that when the class information is unavailable, there is a no effective way to partition the continuous attribute. The contribution of this work towards this challenge is the mode finding which facilitates the discretization of a continuous attribute by maximizing the interdependence between the continuous attribute and the mode. Since the mode is one with the strongest interdependence to the all attributes, it could be functioned as the class label of the attribute clusters. Based on this idea, the proposed mode-driven discretization in which the effectiveness is proven in some of the experiments done transforms the continuous attributes to discrete attributes in a reasonable and systematic manner for further pattern analysis.

6.1.3 Application Contributions

1. Unsupervised discovery of fuzzy pattern of gene expression data [Wu et al. 2010]. Discovering patterns from gene expression levels is regarded as a classification problem when tissue classes of the samples are given and solved as a discrete-data problem by discretizing the expression levels of each gene into intervals maximizing the interdependence between that gene and the class labels. However, when class information is unavailable, discovering gene expression patterns becomes difficult. For a gene pool with a large number of genes, we first cluster the genes into smaller groups. In each group, we use the representative gene, one with highest interdependence with others in the group, to drive the discretization of the gene expression levels of other genes. Treating intervals as discrete events, association patterns can be discovered. If the gene groups obtained are crisp clusters, significant patterns overlapping different clusters cannot be found. Based

on the concept of "fuzzifying" the crisp attribute clusters, we detect patterns overlapping different gene groups. In our experiment using a gene expression dataset with known class labels, we analyze it without relying on the class labels but used later as the ground truth in a classificatory problem for assessing the algorithm's effectiveness in fuzzy gene clustering and discretization. The results show the efficacy of the proposed method.

2. Discovery and grouping of meteorological patterns from surface stations over a large area rendering subtle information for regional weather monitoring [Wong et al. 2010].

   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. Discovery and grouping of parameter patterns in delay coking process revealing system function and operational characteristics [Wong et al. 2010].

   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.

6.2 Suggestions for Further Research

This thesis has developed a unified framework for discovering patterns for sequence and mixed-mode data. In the future, we are going to generalize our method to handle more types of to arrive at an integrated prototype for researchers and general users. Some suggested future research is as follows.

1) After solving the problem of a) the class labels unavailability and b) unsupervised discretization problem, the related technology developed for pattern clustering can now be on categorical data only, which is not able to apply to sequence, continuous and mixed-mode data. Thus, as a natural extension work of this research, it is in need to integrate the proposed system with the related technology including pattern clustering, summarization and visualization.

2) The successful experiments to produce insightful patterns and solutions to two difficult real world problems with large databases encourage the current work to apply to other large mixed-mode database for pattern discovery and data mining. By investigating both the attribute subgroups and the patterns in the perspective of the application domain, it is anticipated that the next step is to generate models and knowledge for further exploration of the new data.

3) Being an ultimate goal, the development of an integrated data mining system for pattern discovery, pattern clustering, summarization and visualization system for generic data with/without class information using possible alternatives is worth to explore.

References

[1] Agrawal R., and Srikant R. "Fast Algorithms for Mining Association Rules", In *Proceedings of the 20th International Conference on Very Large Data Bases*, Santiago de Chile, Chile, 487-499, (September) 1994.

[2] Agrawal R., Gehrke J., Gunopulos D., and Raghavan P. "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications", In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Seattle, Washington, 94-105, (June) 1998.

[3] Agrawal R., Ghost S., Imielinski T., Iyer B., and Swami A. "An Interval Classifier for Database Mining Applications", In *Proceedings of the 18th International Conference on Very Large Data Bases*, Vancouver, British Columbia, Canada, 560-573, (August) 1992.

[4] Agrawal R., Imielinski T., and Swami A. "Mining Association Rules between Sets of Items in Large Databases", In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Washington D.C., USA, 207-216, (June) 1993.

[5] Aleksander I. and Morton H. "Introduction to Neural Computing", North Oxford Press, 1990.

[6] Alon U., Barkai N., Notterman D.A., Gish K., Ybarra S., Mack D., and Levine A.J. "Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays", In *Proceedings of the National Academy of Sciences of the United States of America*, 96, 6745-6750, 1999.

[7] Asuncion A., and Newman D.J. "UCI Machine Learning Repository", School of Information and Computer Science, University of California, Irvine, California. 2007. DOI = http://archive.ics.uci.edu/ml/.

[8] Asyali M.H., Alci M. "Reliability Analysis of Microarray Data using Fuzzy C-Means and Normal Mixture Modeling Based Classification Methods", Bioinformatics, 21, 5, 644-649, 2005.

[9] Au W.H., Chan K.C.C., Wong A.K.C., and Wang Y. "Attribute Clustering for Grouping, Selection, And Classification of Gene Expression Data", IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2, 2, 83-101, (April-June) 2005.

[10] Baraldi A., and Blonda P. "A Survey of Fuzzy Clustering Algorithms for Pattern Recognition - Part I", IEEE Transactions on Systems, Man, and Cybernetics - Part B, 29, 6, 778-785, 1999.

[11] Beale R., and Jackson T. "Neural Computing - An Introduction", Adam Hilger, Bristol, England, 1990.

[12] Belacel N., Cuperlovic-Culf M., Laflamme M., and Ouelette R. "Fuzzy J-Means and VNS Methods for Clustering Genes from Microarray Data", Bioinformatics, 20, 11, 1690-1701, 2004.

[13] Berkhin P. "Survey of Clustering Data mining Techniques", Technical Report, Accrue Software, San Jose, California, 2002.

[14] Bezdek J., Keller J., Krishnapuran R., and Pal N. "Fuzzy Models and Algorithms for Pattern Recognition and Image Processing", Kluwer Academic Publishers, Norwell, Massachusetts, 1999.

[15] Breiman L., Friedman J.H., Olshen R.A. and Stone C.J. "Classification and Regression Trees", Wadsworth International, Belmont, California, 1984.

[16] Brin S., Motwani R., and Silverstein C. "Beyond Market Baskets: Generalizing Association Rules to Correlations", In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Tucson, Arizona, 265-276, (May) 1997.

[17] Brin S., Motwani R., Ullman J.D., and Tsur S. "Dynamic Itemset Counting and Implication Rules for Market Basket Data", In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Tucson, Arizona, 255-264, 1997.

[18] Cadez I., Heckerman D., Meek C., Smyth P., and White S. "Model-based Clustering and Visualization of Navigation Patterns on a Web Site", Data Mining and Knowledge Discovery, 7, 4, 399-424, 2003.

[19] Chan K.C.C., and Wong A.K.C. "APACS: A System for the Automatic Analysis and Classification of Conceptual Patterns", Computational Intelligence, 6, 3, 119-131, 1990.

[20] Chan K.C.C., and Wong A.K.C. "Statistical Technique for Extracting Classificatory Knowledge from Databases", Knowledge Discovery in Databases. AAAI/MIT Press, 107-123, 1991.

[21] Chan K.C.C., Ching J.Y., and Wong A.K.C. "A Probabilistic Inductive Learning Approach to the Acquisition of Knowledge in Medical Expert Systems", In *Proceedings of the Fifth Annual IEEE Symposium Computer-Based Medical Systems*, Durham, North Carolina, 572-281, 1992.

[22] Chau T., and Wong A.K.C. "Pattern Discovery by Residual Analysis and Recursive Partitioning", IEEE Transactions on Knowledge and Data Engineering, 11, 6, 833-854, (November) 1999.

[23] Cheng Y., and Church G.M. “Biclustering of Expression Data”, In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, San Diego, California, 93-103, 2000.

[24] Cheung D.W., Ng V., Fu A.W. and Fu Y. "Efficient Mining of Association Rules in Distributed Databases", Special Issue in Data Mining, IEEE Transactions on Knowledge and Data Engineering, 8, 6, 911-922, 1996.

[25] Ching J.Y., Wong A.K.C., and Chan K.C.C. "Class-Dependent Discretization for Inductive Learning From Continuous And Mixed-Mode Data", IEEE Transactions on Pattern Analysis and Machine Intelligence, 17, 7, 641-651, 1995.

[26] Chiu D.K.Y., Wong A.K.C., and Cheung B. "Information Discovery through Hierarchical Maximum Entropy Discretization and Synthesis", Knowledge Discovery in Databases, AAAI Press, 125-140, 1991.

[27] Cios K.J., and Kurgan L.A. "CLIP4: Hybrid Inductive Machine Learning Algorithm That Generates Inequality Rules", Information Science, 163, 1-3, 37-83, 2004.

[28] Cios K.J., and Kurgan L.A. "Hybrid Inductive Machine Learning: An Overview of Clip Algorithms", New Learning Paradigms in Soft Computing, Physica-Verlag (Springer), 276-322, 2001.

[29] Clark P., and Niblett T. "The CN2 Algorithm", Machine Learning, 3, 4, 261-283, 1989.

[30] Cordero F., Botta M., and Calogero R.A. "Microarray Data Analysis and Mining Approaches", Briefings in Functional Genomics and Proteomics, 6, 4, 265-81, 2008.

[31] Dasarathy B. "Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques", IEEE Computer Society Press, Los Alamitos, California, 1991.

[32] Davis L. "Handbook of Genetic Algorithms", Academic Press, New York, 1991.

[33] Dempster P. "Upper and Lower Probabilities Induced by a Multi-Valued Mapping", Annals of Mathematical Statistics, 38, 325-339, 1967.

[34] Dias J.G., and Cortinhal M.J. "The SKM Algorithm: a *k*-Means Algorithm for Clustering Sequential Data", Advances in Artificial Intelligence - IBERAMIA, Springer Berlin Heidelberg, 173-182, 2008.

[35] Dias J.G., and Vermunt J.K. "Latent Class Modeling of Website Users' Search Patterns: Implications for Online Market Segmentation", Journal of Retailing and Consumer Services, 14, 4, 359-368, 2007.

[36] Ding C., and Peng H. "Minimum Redundancy Feature Selection from Microarray Gene Expression Data", In *Proceedings of the IEEE Computational Systems Bioinformatics Conference*, Stanford, California, 523-528, 2003.

[37] Domany E. "Cluster Analysis of Gene Expression Data", Journal of Statistical Physics, 110, 1117-1139, 2003.

[38] Economist T. "A Special Report on Managing Information: Data, Data Everywhere", The Economist, (February 25th) 2010.

[39] Eidhammer I. "Protein Bioinformatics: An Algorithmic Approach to Sequence and Structure Analysis", John Wiley & Sons, Chichester, UK, 2004.

[40] Eisen M.B., Spellman P.T., Brown P.O., and Botstein D. "Cluster Analysis and Display of Genome-Wide Expression Patterns", In *Proceedings of the National Academy of Sciences of the United States of America*, 95, 25, 14863-14868, 1998.

[41] Fayyad U.M., and Irani K.B. "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning", In *Proceeding of Thirteenth International Conference on Artificial Intelligence*, 1022-1027, 1993.

[42] Fern X.Z., and Brodley C.E. "Random Projection for High Dimensional Data Clustering: a Cluster Ensemble Approach", In *Proceedings of 20th International Conference on Machine Learning*, Washington D.C, 186-193, 2003.

[43] Forgy E. "Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classifications", Biometrics, 21, 768-769, 1965.

[44] Gasch A.P., and Eisen M.B. "Exploring the Conditional Coregulation of Yeast Gene Expression through Fuzzy *k*-means Clustering", Genome Biology, 3, 11, 0059.1-0059.22, 2002.

[45] Goldberg D.E. "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, Reading, Massachusetts, 1989.

[46] Haberman S.J. "The Analysis of Residuals in Cross-Classified Tables", Biometrics, 29, 205-220, 1973.

[47] Han J., and Kamber M. "Data Mining: Concepts and Techniques", Morgan Kaufmann, San Francisco, 2001.

[48] Heyer L. J., Kruglyak S., and Yooseph S. "Exploring Expression Data: Identification and Analysis of Coexpressed Genes", Genome Research, 9, 1106-1115, 1999.

[49] Holland J.H. "Genetic Algorithms and Classifier Systems: Foundations and Future Directions", In *Proceedings of 2nd International Conference on Genetic Algorithms on Genetic Algorithms and Their Applications*, 82-89, 1987.

[50] Houtsma M., and Swami A. "Set-oriented Mining of Association Rules", Research Report RJ 9567, IBM Almaden Research Center, San Jose, California, 1993.

[51] Jain A. K., Murty M.N., and Flynn, P. J. "Data Clustering: A Review", ACM Computing Surveys, 31, 3, 264-323, 1999.

[52] Jiang D., Tang C., and Zhang A. "Cluster Analysis for Gene Expression Data: A Survey", IEEE Transactions on Knowledge and Data Engineering, 16, 11, 1370-1386, 2004.

[53] Karypis G., Han E.H., and Kumar V. "CHAMELEON: a Hierarchical Clustering Algorithm using Dynamic Modeling", Computer, 32, 68-75, 1999.

[54] Kaufman K.A., and Michalski R.S. "Learning from Inconsistent and Noisy Data: The AQ18 Approach", In *Proceedings of the 11th International Symposium on Foundations of Intelligent Systems*, 411-419, (June) 1999.

[55] Kaufman L., and Rousseeuw, P.J. "Finding Groups in Data: an Introduction to Cluster Analysis", John Wiley, New York, 1990.

[56] Kerber R. "ChiMerge: Discretization of Numeric Attributes", In *Proceeding of Ninth International Conference on Artificial Intelligence*, 123-128, 1992.

[57] Kohonen T. "Self-Organization and Associative Memory", Springer-Verlag, New York, 1989.

[58] Kurgan L., and Cios K.J. "CAIM Discretization Algorithm", IEEE Transactions on Knowledge and Data Engineering, 16, 2, 145-153, 2004.

[59] Kurgan L., and Cios K.J. "Fast Class-Attribute Interdependence Maximization (CAIM) Discretization Algorithm", In *Proceedings of the 20th International Conference on Machine Learning and Applications*, Washington D.C, 30-36, 2003.

[60] Lewin B. "Genes VIII", Pearson Education, Inc., New York, 25-28, 2004.

[61] Liu H., and Setiono R. "Feature Selection via Discretization", IEEE Transactions on Knowledge and Data Engineering, 9, 4, 642-645, 1997.

[62] Liu H., Hussain F., Tan C.L., and Dash M. "Discretization: an Enabling Technique", Journal of Data Mining and Knowledge Discovery, 6, 4, 393-423, 2002.

[63] Liu L., Wong A.K.C., and Wang Y. "A Global Optimal Algorithm for Class-Dependent Discretization of Continuous Data", Intelligent Data Analysis, 8, 2, 151-170, 2004.

[64] Ma P.C.H., and Chan K.C.C. "UPSEC: an Algorithm for Classifying Unaligned Protein Sequences into Functional Families", Journal of Computational Biology, 15, 4, 431-443, 2008.

[65] Ma P.C.H., Chan K.C.C., and Chiu D.K.Y. "Clustering and Re-Clustering for Pattern Discovery in Gene Expression Data", Journal of Bioinformatics and Computational Biology, 3, 2, 281-301, 2005.

[66] MacQueen J.B. "Some Methods for Classification and Analysis of Multivariate Observations", In *Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281-297, 1967.

[67] Madeira S.C., and Oliveira A.L. “Biclustering Algorithms for Biological Data Analysis: A Survey", IEEE Transactions on Computational Biology and Bioinformatics, 1, 1, 24-45, 2004.

[68] Michalski R.S., Mozetic I., Hong J., and Lavrac N. "The Multi-Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical domains", In *Proceedings of Fifth National Conference on Artificial Intelligence*, Philadelphia, Pennsylvania, 1041-1045, 1986.

[69] Milligan G. "An Examination of the Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms", Psychometrica, 45, 325-342, 1980.

[70] Mitra P., Murthy C.A., Pal S.K. “Unsupervised Feature Selection Using Feature Similarity”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24, 4, 301-312, 2002.

[71] Ng R., and Han J. "Efficient and Effective Clustering Methods for Spatial Data Mining", In *Proceedings of the 20th International Conference on Very Large Data Bases*, Santiago de Chile, Chile, 144-155, (September) 1994.

[72] Osteyee D.B., and Good I.J. "Information, Weight of Evidence, the Singularity between Probability Measures and Signal Detection", Springer-Velag, Berlin, Germany, 1974.

[73] Parsons L., Haque E., and Liu H. “Subspace Clustering for High Dimensional Data: A Review”, ACM SIGKDD Explorations Newsletter, 6, 1, 90-105, 2004.

[74] Paterson A., and Niblett T.B. "ACLS Manual", Intelligent Terminals Ltd, Edinburgh, Scottland, 1987.

[75] Pease A.C., Solas D., Sullivan E.J., Cronin M.T., Holmes C.P., Fodor S.P. "Light-generated Oligonucleotide Arrays for Rapid DNA Sequence Analysis", In *Proceedings of the National Academy of Sciences of the United States of America*, 91, 11, 5022-5026, (May) 1994.

[76] Pedrycz W., and Gomide F. "An Introduction to Fuzzy Sets: Analysis and Design", The MIT Press, Cambridge, Massachusetts, 1998.

[77] Pevsner J. "Bioinformatics and Functional Genomics", Wiley-Liss, Inc., Hoboken, New Jersey, 2003.

[78] Piatetsky-Shapiro, Khabaza T., and Ramaswamy S. "Capturing Best Practice for Microarray Gene Expression Data Analysis", In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington D.C., 407-415, 2003.

[79] Quinlan J.R. "C4.5: Programs for Machine Learning", Morgan Kaufmann, San Mateo, California, 1993.

[80] Quinlan J.R. "Generating Production Rules from Decision Tress", In *Proceedings of the International Joint Conference on Artificial Intelligence*, 304-307, 1987.

[81] Ruping S. "mySVM Manual", University of Dortmund. Lehrstuhl Informatik 8. 2000. DOI = http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/.

[82] Schena M., Shalon D., Davis R.W. and Brown P.O. "Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray", Science, 270, 5235, 467-470, (October) 1995.

[83] SCPD, The Promoter Database of Saccharomyces cerevisiae. 2010. DOI= http://rulai.cshl.edu/SCPD/.

[84] Smet F.D., Mathys J., Marchal K., Thijs G., Moor B.D., and Moreau Y. “Adaptive Quality-Based Clustering of Gene Expression Profiles", Bioinformatics, 18, 5, 735-746, 2002.

[85] Smyth P., and Goodman R.M. "An Information Theoretic Approach to Rule Induction from Databases", IEEE Transactions on Knowledge and Data Engineering, 4, 4, 301-316, 1992.

[86] Smyth P., Fayyad U., Piatetsy-Shapiro, and Uthurusamy R. "Advances in Knowledge Discovery and Data Mining", MIT Press, Cambridge, Massachusetts, 1996.

[87] Su C.T., and Hsu J.H. "An Extended Chi2 Algorithm for Discretization of Real Value Attributes", IEEE Transactions on Knowledge and Data Engineering, 17, 3, 437-441, 2005.

[88] Tamayo P., Solni D., Mesirov J., Zhu Q., kitareewan S., Dmitrovsky E., Lander E.S., and Golub T.R. "Interpreting Patterns of Gene Expression with Self-Organizing Maps: Methods and Application to Hematopoietic Differentiation", In *Proceedings of the National Academy of Sciences of the United States of America*, 96, 6, 2907-2912, 1999.

[89] Tay F., and Shen L. "A Modified Chi2 Algorithm for Discretization", IEEE Transactions on Knowledge and Data Engineering, 14, 3, 666-670, 2002.

[90] Tou J.T., and Gonzalez R.C. "Pattern Recognition Principles", Addison-Wesley, London, 1974.

[91] Tsai. C.J., Lee C.I., Yang W.P. "A Discretization Algorithm Based on Class-Attribute Contingency Coefficient", Information Sciences, 178, 3, 714-731, (February) 2008.

[92] Wang C.C., and Wong A.K.C. "Classification of Discrete-Valued Data with Feature Space Transformation", IEEE Transactions on Automatic Control, 24, 3, 434-437, 1979.

[93] Wang J., Bo T.H., Jonassen I., Myklebost O., and Hovig E. "Tumor Classification and Marker Prediction by Feature Selection and Fuzzy *c*-means Clustering using Microarray Data", BMC Bioinformatics, 4, 60, (December) 2003.

[94] Wang Y., and Wong A.K.C. "Discover*e", Pattern Discovery Technologies, 2010. DOI= http://www.patterndiscovery.com.

[95] Wang Y., and Wong A.K.C. "From Association to Classification: Inference using Weight of Evidence", IEEE Transactions on Knowledge and Data Engineering, 15, 3, 764-767, 2003.

[96] Wong A.K.C., and Chiu D.K.Y. “Synthesizing Statistical Knowledge from Incomplete Mixed-Mode Data”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 9, 8, 796-805, 1987.

[97] Wong A.K.C., and Li G.C.L. "Association Pattern Analysis for Pattern Pruning, Pattern Clustering and Summarization", To appear in Journal of Knowledge and Information Systems, 2010.

[98] Wong A.K.C., and Li G.C.L. "Simultaneous Pattern and Data Clustering for Pattern Cluster Analysis", IEEE Transactions Knowledge Data Engineering, 20, 7, 911-923, 2008.

[99] Wong A.K.C., and Liu T.S. "Typicality, Diversity and Feature Patterns of an Ensemble", IEEE Transactions on Computers, 24, 2, 158-181, 1975.

[100] Wong A.K.C., and Wang C.C. "DECA - a Discrete-Valued Data Clustering Algorithm", IEEE Transactions on Pattern Analysis and Machine Intelligence, 1, 4, 342-349, 1979.

[101] Wong A.K.C., and Wang Y. "High Order Pattern Discovery from Discrete-Valued Data", IEEE Transactions on Knowledge and Data Engineering, 9, 6, 877-893, 1997.

[102] Wong A.K.C., and Wang Y. "Pattern Discovery: a Data Driven Approach to Decision Support", IEEE Transactions on Systems, Man, and Cybernetics - Part C, 33, 1, 114-124, 2003.

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

[104] Wong A.K.C., Wu B., Wu G.P.K., and Chan K.C.C. "Pattern Discovery for Large Mixed Mode Database", In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, Toronto, Canada, 859-868, (October) 2010.

[105] Wu B. "Sinocan Intellitech Ltd", 2010. DOI= http://www.sinocansystems.com.cn.

[106] Wu G.P.K., Chan K.C.C., Wong A.K.C., and Wu B. "Unsupervised Discovery of Fuzzy Patterns in Gene Expression Data", To appear in *Proceedings of the IEEE International Conference on Bioinformatics & Biomedicine*, Hong Kong, (December) 2010.

[107] Wu Q., Bell D.A., McGinnity T.M., Prasad G., Qi G., and Huang X. "Improvement of Decision Accuracy using Discretization of Continuous Attributes",

In *Proceedings of the Third International Conference on Fuzzy Systems and Knowledge Discovery*, Lecture Notes in Computer Science, 4223, 674-683, 2006.

[108] Yen J., and Langari R. "Fuzzy Logic: Intelligence, Control, and Information", Upper Saddle River, Prentice-Hall, New Jersey, 1999.

[109] Yu L., and Liu H. "Redundancy Based Feature Selection for Microarray Data", In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, Washington, 737-742, 2004.

[110] Zhang A. "Advanced Analysis of Gene Expression Microarray Data", World Scientific, 2006.

[111] Zhang T., Ramakrishnan R., and Livny M. "BIRCH: an Efficient Data Clustering Method for Very Large Databases", In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 103-114, Montreal, Quebec, Canada, (June) 1996.

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