DOMAIN-CONCEPT MINING:

AN EFFICIENT ON-DEMAND DATA MINING APPROACH

A Dissertation

presented to

the Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

---

by

WANNAPA KAY MAHAMANEERAT

Dr. Chi-Ren Shyu, Dissertation Supervisor

AUGUST 2008

The undersigned, appointed by the dean of the Graduate School, have examined the dissertation entitled

DOMAIN-CONCEPT MINING: AN EFFICIENT ON-DEMAND DATA MINING APPROACH

presented by Wannapa Mahamaneerat,

a candidate for the degree of doctor of philosophy,

and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Chi-Ren Shyu

Dr. Dong Xu

Dr. Guilherme N. DeSouza

Dr. C. Alec Chang

Dr. Jane M. Armer

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my advisor, Dr. Chi-Ren Shyu, for his continuous support. His dedication, invaluable guidance, and selfless devotion throughout the development of this dissertation are second to none. I would like to thank the doctoral committee: Dr. Dong Xu and Dr. Guilherme N. DeSouza for providing fruitful discussion for my research, and Dr. C. Alec Chang, who co-authored my first journal article, for his expertise he shared with me. A special thanks is expressed to another doctoral committee and my around-the-world travel partner, Dr. Jane M. Armer. Her positive attitude and encouragement have enabled me to reach my best potential.

A thanks also goes to Dr. Fungai Chanetsa, who introduced me to a challenging public health data set, which inspired the creation of Domain-Concept Mining (DCM). An appreciation is also expressed to the MedBio lab members for their honest comments, which have shaped DCM well. In addition, my gratitude is expressed specifically to Mr. Tetsuya Kobayashi and Mr. Jason M. Green who have made DCM successful in a competition.

I would like to thank my father, Mr. Sunthorn Mahamaneerat, and my family for their patience and unconditional support. A thanks also goes to Mrs. Chattavee Numtee-Maloney, my wonderful friend, who has tirelessly proofread this dissertation. Last but not least, a thanks must go to my fiancé, Mr. Steve J. Apperson, who genuinely loves what he does every minute of the day. The strength I have gained from witnessing his work contributes to the completion of this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

DOMAIN-CONCEPT MINING: AN EFFICIENT ON-DEMAND DATA MINING APPROACH

Wannapa Kay Mahamaneerat
Dr. Chi-Ren Shyu, Dissertation Supervisor

ABSTRACT

Traditional brute-force association mining approaches, when applied to large datasets, are thorough but inefficient due to computational complexity. A low global minimum probability threshold can worsen this complexity by producing an overwhelming number of associations; however, a high threshold may not uncover valuable associations, especially from under-represented groups within the population. Regardless, the uncovered associations are not systematically organized.

To solve these problems, novel Domain-Concept Mining (DCM) with Partition Aggregation (DCM-PA) has been developed. DCM organizes data by grouping transactions with common characteristics, such as a certain age group, into "domain-concepts" (dc). DCM granulizes partitioning criteria by pairing each attribute with its values. Criteria may include under-represented groups as well as spatial, temporal, and incremental dimensionalities. Then, a statistical power analysis is utilized to determine if multiple criteria of the same attribute, such as age group 18-24 and 25-34, should be combined to form a broader partition. Doing so maintains the tradeoff between findings with statistical significance and computational resource consumptions, while preserving data organization. Associations can be extracted from each partition independently because a partition contains all of its qualified transactions. Moreover, the partition size proportionally adjusts the global threshold to be more specific and sensitive.

After the initial phase is complete, DCM-PA efficiently reuses DCM's associations to compute results from multiple-partition aggregation (union or intersection) using Bayes Theorem and a pipelining technique. DCM-PA offers the flexibility to perform association mining that is expected to uncovering more valuable knowledge through means like trends and comparisons from various dc partitions and their aggregations.

CHAPTER 1

INTRODUCTION

1.1 Motivations

Descriptive data mining, or *association mining*, may be understood as an analysis tool intended to discover *knowledge*, which can be defined as logical associations or relationships, homogenous patterns, and/or hidden correlations among data attributes or variables [1-4]. Some association mining approaches may further evaluate discovered knowledge by discerning whether it is consistent with previously uncovered knowledge or it is in fact a novel item, which may suggest additional studies [5-7]. With the rapid and incessant growth of the data collected [1, 8-10], users, which include human subject-matter experts who may own or collect the data, have encountered difficulties in finding ways to comprehend and utilize the data directly without some forms of data aggregations, summarizations, and analyses [11]. So far, association mining [12-18] has been known to be one of the most successful approaches in order to respond to the ever growing need of data management [9, 19]. Unfortunately, traditional brute-force association mining approaches, which are based on the association rule (AR) mining that was originally proposed by Agrawal et al. [12], have been shown not to be efficient enough to mine very large data sets entirely, due to the amount of memory required for

its iterative comparison processes [20, 21]. Another problem with the traditional brute-force mining approaches is that the overwhelming amount of the mining results [21, 22] is unorganized [23, 24], and hard to browse or search [23, 25]. This problem is caused by an assumption of mining with no prior hypothesis, with a purpose of discovering the complete set of associations [22, 23]. Hence, the association mining effort and its uncovered knowledge may not be utilized to their potential because users end up with similar difficulties in comprehending extremely large amounts of information [7, 24].

An important factor of the association mining problems is its global minimum probability threshold [5, 26], called "minimum support". An association that is uncovered from a data set is determined by comparing the probability (proportion, occurrence frequency [20], or percentage of transactions [27]) of its co-occurring attributes with the minimum support threshold [12, 13, 27, 28]. When the minimum support is low, the association mining approach may uncover valuable associations or knowledge [26, 27], which may be associated with under-represented groups of population [24]. However, lowering the minimum support value increases the computational complexity [27] because millions of associations [29], whether they are valuable or trivial, may be reported. On the contrary, raising the minimum support, which reduces the amount of the computational resources required, may result in an inability to discover valuable knowledge from the data [27].

In practical settings, human experts may apply their experience and form expectations of what a valuable finding from the data would be [25]. Many valuable findings are usually non-trivial [30], which implies that the finding are from co-occurring attributes that have low probabilities. Hence, uncovering such valuable findings directly

from the entire data set may not be efficient, because an association mining approach likely uncovers many trivial associations than non-trivial ones. Further, identifying non-trivial associations from a pool of associations will lead to a double-effort of the association mining [22, 24].

1.2 A Need for Domain-Concept Mining

To solve the afore-mentioned problems, we have developed a novel data mining approach, called *Domain-Concept Mining* (DCM), to: 1.) uncover non-trivial findings directly, 2.) improve mining efficiency by reducing the size of the data set, and 3.) organize the resulting associations using data characteristics. DCM first organizes the data before analysis through its unique partitioning technique by grouping transactions that share some common characteristics together. For example, all transactions with the same age group ranging from 18 to 24 will be in the same partition. The data characteristics are drawn systematically by pairing each attribute (or variable) with its values. DCM granulizes a characteristic of the data to be (one *attribute*: one *value*) pair, called "domain-concept" (dc). This step is done for all (*attribute*: *value*) pairs, which also include the under-represented groups' characteristics, and temporal, spatial, and incremental dimensionalities of the data when available. All domain-concepts are considered as DCM's potential partitioning criterion. The purposes of partitioning the data are: 1.) to increase the efficiency of the association mining by reducing the number of transactions for each partition, 2.) to include all qualified transactions according to the partition's criterion into a partition, so the mining step can be done for each partition

independently, and 3.) to naturally organize the data according to their characteristics. It is worth mentioning that the mining results will also be organized accordingly.

However, a tradeoff between increasing the association mining efficiency and the findings with statistical significance should also be maintained. DCM utilizes a statistical power analysis (also called a sample size estimation) [31, 32], which recommends the minimum number of transactions for a partition. When necessary, DCM combines two or more domain-concepts with the same attribute to form a broader domain-concept. For example, if too few transactions qualify for the (*age group*: *18-24*) domain-concept, DCM broaden this domain-concept by combining it with (*age group*: *25-34*), and so on, until the recommended number of transactions is achieved. Doing so balances the tradeoff and also preserves the organization of the partitions.

After the transactions are grouped into domain-concept partitions, DCM implements an association mining algorithm, called "Frequent Pattern Tree" (FPT or FP-Tree) [20], to uncover findings from each partition separately and independently in batches of distributed or parallel fashion. DCM uses one global minimum support threshold, which is automatically and locally adjusted according to the partition size; hence, the threshold can be more sensitive and specific to the distribution of the domain-concept.

It is worth mentioning that dc partitions, which represent characteristics of the data, may have different sizes. Since DCM uses all attributes to partition the data, it is possible that some dc partitions may overlap each other. To better explain the unique features of the afore-mentioned dc partitions and the DCM partitioning technique, a generic example of a biomedical informatics data set is used (as shown in Table 1.1).

TIDs represent a transaction identifiers, which are numeric values from 1 to *n*, where *n* is the total number of transactions in the entire data set. Columns *A* to *E* represent (*attribute*: *value*) pairs of (*chronic disease*: *diabetes*), (*healthcare coverage*: *no*), (*age group*: *65-74*), (*diagnosis*: *stroke*), and (*high blood pressure*: *yes*), respectively. Please note that an attribute, such as *chronic disease*, may have many other possible values, such as heart disease and cancer. The attribute values shown in Table 1.1 are scoped down for simplicity. To further simplify the explanation, variables *A* to *E* may be used for each corresponding (*attribute*: *value*) pair.

Table 1.1. A Generic Example of a Biomedical Informatics Data Set

TID	*A* (*chronic disease*: *diabetes*)	*B* (*healthcare coverage*: *no*)	*C* (*age group*: *65-74*)	*D* (*stroke*: *yes*)	*E* (*high blood pressure*: *yes*)
1	√	√	√		
2	√	√			√
3	√	√	√	√	
4		√	√		√
.		√			√
.	√		√		√
.				√	
n-1	√				√
n			√	√	√

Let us assume that each domain-concept is a set of (*attribute*: *value*) pair, and the statistic power analysis does not suggest any combination of domain-concepts. In this setting, DCM partitions the data into five dc partitions, *A*, *B*, *C*, *D*, and *E*, where the transactions that share (*chronic disease*: *diabetes*) are in partition *A*, and so forth.

Moreover, the dc partition *A* may have a different number of transactions from the dc partition *B*, depending on the actual qualified transactions.

It is important to mention that dc partitions are based on actual distributions of data. Furthermore, multiple (*attribute*: *value*) pairs make up transactions. Hence, the resulting dc partitions may overlap each other, (e.g. TIDs 2 and 3 are the overlapping transactions between partitions *A* and *B*).

1.2.1 A Real-World Example

To further understand the benefit from the DCM partitioning technique that groups transactions according to their common (*attribute*: *value*) pair(s), Figure 1.1 illustrates the distributions of an attribute "*Would you say that in general your health is?*" (or *GENHLTH*). Figure 1.1 (a) shows the distribution of *GENHLTH*'s values: excellent, very good, good, fair, and poor, and (b) shows the distribution of *GENHLTH*'s values among those who have been told by a doctor they have diabetes (*DIABETES*: *yes*). All values shown in Figure 1.1 are drawn from a public health data set of the "Centers for Disease Control and Prevention" (CDC), called the "Behavioral Risk Factor Surveillance System" (BRFSS) 2006 [33]. In addition, the entire BRFSS 2006 has 355,710 transactions.

Suppose a human expert is interested in uncovering associations that are related to (*DIABETES*: *yes*) and other health risks, behaviors, or factors. Findings such as associations between (*DIABETES*: *no*) and either (*GENHLTH*: *excellent*), (*GENHLTH*: *very good*), or (*GENHLTH*: *good*) are considered trivial and not of the expert's interest. As shown in Figure 1.1 (b), only 18.1% of all transactions that have (*DIABETES*: *yes*) are

associated with (*GENHLTH*: *poor*), which is only 6,480 transactions (or 1.8%) of the entire BRFSS 2006 data set.

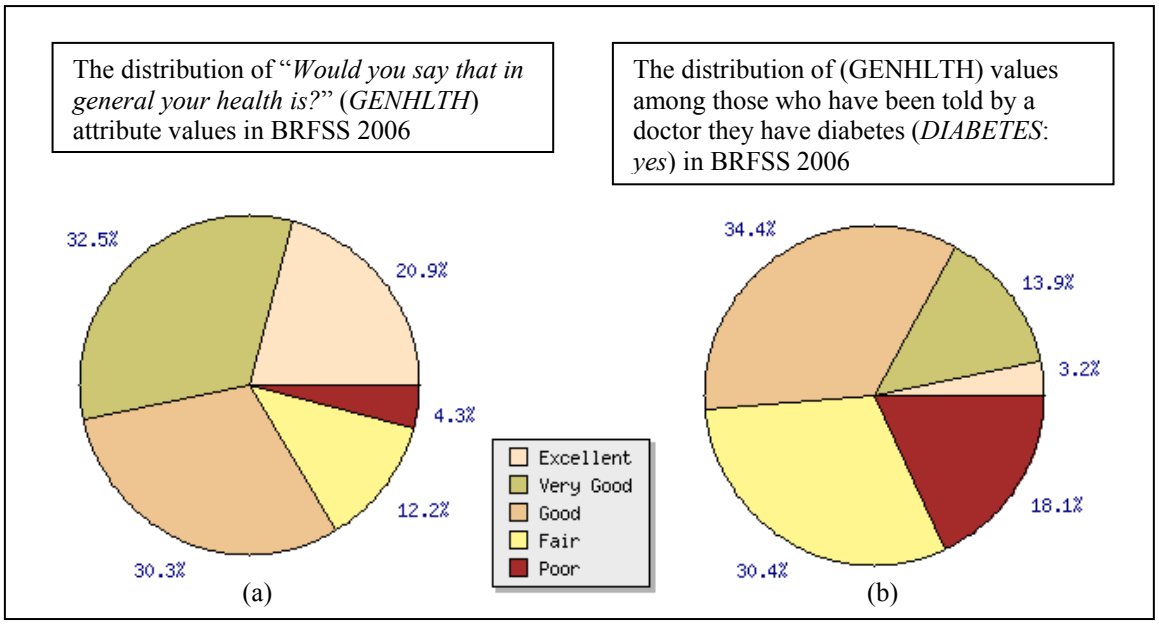


Figure 1.1. An example of (a) attribute values and their distributions in comparison to (b) the same attribute values, but different distributions once another variable is taken into considerations.

Without the DCM partitioning technique, which groups all transactions that have (*DIABETES*: *yes*) as a partition, one may have to set the minimum support threshold, lower than 1.8% to uncover any associations beyond (*DIABETES*: *yes*) and (*GENHLTH*: *poor*). For example, an association that contains (*DIABETES*: *yes*), (*GENHLTH*: *poor*), and (*QLACTLM*: *yes*) (where *QLACTLM* is "Are you limited in any way in any activities because of physical, mental, or emotional problems?") has only 4,969 qualified transactions (or 1.4% of 355,170 transactions). By lowering the support threshold in order to uncover such association may cause a brute-force association mining approach to report an overwhelming amount of trivial associations. This scenario may also cause a memory exhaustion problem because the complete (and redundant subsets) of

associations would be reported. Mining efficiency may be increased by reducing the number of transactions, which will be detailed when DCM is discussed in Chapter 3 Section 3.3.3. The discussion in the same section will also detail how DCM uses the statistical power analysis sample size estimation in maintaining the minimum number of transactions of a dc partition.

Since characteristics of the under-represented groups of data are their (*attribute*: *value*) pairs, these characteristics can also be designated as dc's to be used as partitioning criteria so that DCM can uncover associations from these dc partitions directly. Moreover, the associations from all dc partitions are organized according to their dc's. Result organizations facilitate users when they view or search the associations. More importantly, trends and comparisons of the associations from various dc partitions can be built to expand the usefulness and understandability of the findings.

Moreover, setting one global minimum support threshold (e.g. 0.1 or 10%) for all dc partitions has the same effect as adjusting the threshold value according to the actual distribution of an individual characteristic of the data. For example, a DCM mining process reports any associations that co-occur more than 10% of the dc partition (*DIABETES*: *yes*), which has 36,085 transactions. This implies that the minimum support threshold is adjusted to approximately 1% of the entire BRFSS 2006 (355,710 transactions in total). In other words, the DCM partitioning technique introduces a more sensitive and specific minimum support setting that automatically adjusts itself according to the characteristics of the dc partitions and the data distributions.

An extension of the DCM approach, called a "hybrid threshold", is proposed to incorporate statistical analysis to improve the traditional association mining's minimum

support threshold. The purpose is to statistically evaluate (*attribute*: *value*) pairs of an association. The hybrid threshold is a weighted sum of the: 1.) minimum support threshold (to measure the probability of the association), 2.) coefficient of determination value (to measure the strength of the association), and 3.) correlation coefficient (to measure the direction of the correlation). The hybrid threshold has an objective to filter the associations by excluding those associations that may occur frequently, but are not statistically correlated. More detailed discussions and analyses are in Chapter 3.

1.3 A Need for Domain-Concept Mining with On-Demand Partition Aggregation Capabilities

Domain-Concept Mining (DCM) is unique because of its partitioning technique, how it organizes the data mining results, and the support threshold that offers sensitivity and specificity as discussed in the previous section. However, the research main contribution is the ability to allow human experts to flexibly aggregate data mining results on-demand and online by either broadening (union) or narrowing (intersection) the domain-concept (dc) partitions. This is achieved through a novel approach, called Domain-Concept Mining Partition Aggregation (DCM-PA). DCM-PA implements Bayes Theorem [31] in order to intelligently aggregate uncovered associations among various dc partitions by reusing the information obtained during the mining of the associations. Furthermore, DCM-PA incorporates a database query optimization through a pipelining technique [34], which enables DCM-PA to aggregate associations on-demand and on-the-fly without a need to materialize intermediate results.

More importantly, the DCM-PA's ability to union (the transactions of) the dc partitions increase association mining efficiency by eliminating the need to mine the entire data set. Instead, DCM-PA compliments DCM by allowing DCM to mine a set of the domain-concepts that share the same attribute, e.g. mining all partitions of age groups. Then, DCM-PA performs a union operation of these partitions to achieve the complete set of *organized* results as if the entire data set was mined from the ground up. A complete detail of DCM-PA is discussed in Chapter 4.

1.4 A Need for Domain-Concept Mining Online System

To complete the development of DCM and maximize the usefulness of its mining results, a Web system, called "DCMiner," has been developed. DCMiner utilizes the organized results from DCM for users to compare, contrast, and form trends among various dc partitions. DCMiner also offers various result visualization techniques, which includes both tabular and graphical formats. The provided capabilities from DCMiner increase ways of investigating results, which may also lead to discovering other valuable knowledge.

Currently, DCMiner has been implemented to the following data sets: 1.) the "Agency for Healthcare Research and Quality" (AHRQ)'s "Nationwide Inpatient Sample" (NIS) [35], 2.) the "Centers for Disease Control and Prevention" (CDC)'s "Behavioral Risk Factor Surveillance System" (BRFSS) [36], and 3.) the "Callaway Nuclear Power Plant's Action Request" (CAR) data. DCMiner and its details are discussed in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

This chapter contains a discussion of the previously published data mining related articles that have inspired our work. The summarization of the literature reviewed is shown in Table 2.1.

Table 2.1. A Survey Summary of Data Mining Approaches

Approaches	Use Support and Confidence?	Brute- Force or Estimation?	Features
Association Rule Mining			
Mining large itemsets and association rules [12]	Yes	Brute-Force	Large itemsets, association rules mining, syntactic constraints.
Apriori and Candidate Generation algorithms [13]	Yes	Brute-Force	An improvement of the previous approach ([12]) to improve memory efficiency during Candidate Generation Algorithm.
Quantitative association rules mining [28]	Yes	Brute-Force	An extended work of the previous two approaches ([12, 13]) with partitioning of quantitative attributes.
Information-theoretic-lower bound [18]	Yes (support)	Sampling	Build frontier set by using information from previous database scans to reduce the size of candidate set.
Statistical Analysis and Information Theory			
Mutual Information Measure (MIM) [37]	MIM	Brute-Force	Utilize Information Theory in findings association through the entropy and MIM.
Statistics and Data Mining [38]	Yes	Not applicable	Examine statistics and data mining similarities and differences.

Approaches	Use Support and Confidence?	Brute- Force or Estimation?	Features
Correlation Rules (generalizing association rules to correlations) [15, 39]	No	Brute-Force	An implementation of the Chi-square test for independence, monotonically increasing or upward closure property of correlation (the border of correlation).
Regression and Frequent Temporal Patterns of Data Streams (FPT-DS) [40]	Yes	Estimation	Reduce the number of data scan to one and regression-based compact pattern representation.
Quantitative Correlated Patterns (QCPs) approach [41]	No	Brute-Force	Use a statistical correlation analysis and information theory to mine data. Data mining results are not based on frequently co-occurring items, but the items with high Mutual Information Measure values.
Clusters data based on correlation [42]	No	Brute-Force	Utilize data distribution and correlation to cluster and build correlation rules.
Mining rank-correlated sets of numerical attributes [43]	Yes (support)	Brute-Force	Discover patterns combining numerical (using correlation) and categorical attributes.
Bitmap and Granular Computing (Bit-AssoRule) [16]	No	Brute-Force	Use bit operations to find rules based on granular (a set of transactions with same attributes) computing.
Regression Class Mixture Decomposition (RCMD) [44]	No	Estimation	Define inlier attributes that are in *regression class*. The regression class is a subset of the data set that is defined by a regression model.
Subjective measures of interestingness [30]	No	Brute-Force	Propose actionable and unexpected measures based on Bayes rule with positive and negative evidences.
Most interesting rules mining and support/confidence border [45]	Yes	Brute-Force	Propose the *most interesting rules*, which are those that reside along a support/confidence border. In other words, interesting rules are rules that either have high support, high confidence, or both.
Improvement of Frequent Itemset Mining (through Data Structures)			
Frequent Pattern Tree (FP-Tree) approach (Mining frequent patterns without candidate generation) [20, 46, 47]	Yes	Brute-Force	Compact the database to FP-Tree data structure that contains sorted frequent itemsets and build association rules from FP-Tree.

Approaches	Use Support and Confidence?	Brute- Force or Estimation?	Features
PPMCR algorithm and Incrementally Counting Suffix Tree (ICST) [48]	Yes	Estimation	Utilize suffix tree data structure. PPMCR generates a set of patterns using a single database scan then traverse ICST to find non-redundant rules. The approach is based on a statistical correlation analysis.
LOOPBACK and Build Once Mine Once (BOMO) algorithms [27]	No	Brute-Force	Efficiently mine *N k*-itemsets with the highest supports. Build FP-Tree for the longest *k*-itemsets with an assumption the minimum support is set to zero.
Closed Association Rule Mining (CHARM) [49, 50] and other frequent closed itemsets [21, 51, 52]	Yes	Brute-Force	Use itemset lattices and closures to mine frequent 'closed' itemsets, which results in less number of itemsets without a loss of information.
Results Organization			
Direction Setting (DS) rules [22]	Yes	Brute-Force	Prune rules by grouping rules with the same consequent then prune out the rules that do not have correlation between antecedent and consequent.
Multiple support apriori (MSapriori) algorithm [26]	Yes	Brute-Force	Users can specify different minimum support thresholds for different attributes depending on how rare the attributes are in the database. Direction Setting (DS) rules help organized the association rules.
Organize rules based on General-Specific (GS) [23]	Yes	Brute-Force	Propose most-general rules set (MGRS) as top-level rules for users to select and browse for more specific rules
Swap randomization [53]	No	Estimation through randomization	Use randomization and Markov chain approaches to measure significance complex associations and to access the findings.
Association Mining with Partitions and Random Sampling			
BitOp and association rule clustering system (ARCS) [19]	Yes	Brute-Force	Propose two-dimensional association rules clustering – one dimension per attribute.
Direction Setting (DS) rules [22] (also categorized in Results Organization)	Yes	Brute-Force	Prune rules by grouping rules with the same consequent then prune out the rules that do not have correlation between antecedent and consequent.

Approaches	Use Support and Confidence?	Brute- Force or Estimation?	Features
Most interesting rules mining and support/confidence border [45] (also categorized in Statistical Analysis and Information Theory)	Yes	Brute-Force	Propose the most interesting rules are those that reside along a support/confidence border, meaning interesting rules are rules that either have high support, high confidence, or both.
Overall sequential pattern mining, mining by divide-and-conquer [54]	Yes	Brute-Force	Mine sequence patterns, which are composed of sequences of actions.
Regression Class Mixture Decomposition (RCMD) [44]	No	Estimation	Define inlier attributes that are in *regression class*. The regression class is a subset of the data set that is defined by a regression model.
Sliding window filtering (SWF) for incremental mining [55]	Yes	Estimation	Utilize partitions and sliding windows to incrementally mine data by using an un-mined partition to update the already mined results.
Preemptive Distributed Decision Miner (PDDM) and Distributed Dual Decision Miner (DDDM) in Distributed Association Rule Mining (D-ARM) algorithm [56]	Yes	No	Distributed Association Rule Mining with less communication
Binary space partitioning trees and Optimized Gain Rules for numeric attributes mining [57]	Yes	Estimation	Find correlations among one or two numeric attributes and categorized attributes using binary space partitioning trees and dynamic programming.
Data Feature Oriented Data Partition (DFDP) [58]	Yes	Brute-Force	Partition data for parallel processing.
FAST, a novel two-phase sampling based algorithm for discovering association rules [59]	Yes	Sampling	Two-phase sampling.
Incremental Mining			
Regression and Frequent Temporal Patterns of Data Streams (FPT-DS) [40]	Yes	Estimation	One data scan and regression-based compact pattern representation.
Real-time data mining [60]	Not applicable	Not applicable	A review of approaches needed to achieve real-time data mining. The approaches include anomaly detection and data stream mining.

Approaches	Use Support and Confidence?	Brute- Force or Estimation?	Features
Temporal High Utility Itemsets – Mine (THUI-Mine) [61]	No	Estimation	Mine temporal data stream, by including new data and removing obsolete data, based on a measurement, called *utility*.
Sliding window filtering (SWF) for incremental mining [55] (also categorized in Association Mining with Partitions and Random Sampling)	Yes	Estimation	Utilize partitions and sliding windows to incrementally mine data by using an un-mined partition to update the already mined results.
Calendar-based temporal association rule [62]	Yes	Brute-Force	Utilize a user-given calendar schema to mine data for calendar-based temporal association rule. Use a level-based mining.
Statistical Borders for Incremental Mining [63]	Yes, but also with other measurements	Sampling	A novel approach that biases the initial support for patterns mining, but maximizes one of two parameters (precision or recall) for incremental data
Incremental Mining for Temporal Association Rules [64]	Yes (support and "strength")	Estimation	Maintain temporal association rules with numerical attributes and the temporal negative border method [17]. Update rules with new data.
Partial Periodic Patterns in Time-Series Databases [65]	Yes, with adaptation	Brute-Force	Merge rules of two or more databases

2.1 Traditional Association Rule Mining

Data mining has been widely researched for over a decade. Since Agrawal et al. [12] and other closely related work [13, 28] have initiated the association rule (AR) mining field. They proposed a model to mine sets of items from a very large market basket data set specifically used for the following:

1.) co-occurrences of items in the baskets, or called *itemsets*, such as an itemset that contains {*bread*, *butter*, *milk*}, and

2.) association rules (AR) in the form of (*antecedent* → *consequent*), e.g. ({*bread*, *butter*} → *milk*).

Further, they defined that an item is binary. In that a basket of items either contains bread or no bread; hence, two loafs of white bread and one loaf of wheat bread in the same basket means that the bread item exists in the basket.

In 1994, Agrawal and Srikant [13] proposed fast algorithms, "Apriori" and "AprioriTID" algorithms with "apriori-gen" function to mine data sets for associations rules. In general, the Apriori algorithm and apriori-gen function have garnered much attention due to the usefulness of the association rule idea and the difficulty of mining co-occurrences of items. Particularly, the apriori-gen function, which generates candidate sets of *frequent itemsets*, was proposed to speed up the data mining process by eliminating unnecessary database scans. The database scans are used to verify the probability of a set of items in order to determine whether the set is frequent or not. A scan is needed when the size of itemsets grows from l to l+1. Hence, reducing number of items each scan needs to verify will speed up the process.

DCM uses a generalized version of the Apriori algorithm from the binary items. Instead, DCM defines an item to be (*attribute*: *value*) pair, e.g. (*bread*: *white*) is not the same as (*bread*: *wheat*). An item (of size one) is denoted by i_j where j = 1, 2, …, M and M is the total number of (*attribute*: *value*) pairs in a data set.

A set of one or more mutually exclusive items is called an itemset (I^P), where

$$I^P \in \{\{i_j\},\{i_k\},...,\{i_m\},\{i_j,i_k\},...,\{i_j,i_k,...i_m\}\},$$

$j, k, m \in \{1, 2, ..., M\}$, $i \neq j \neq m$, and $i_j \cap i_m = \phi$. Furthermore, an itemset can be called a *frequent* itemset if and only if the joint probability of its co-occurring items is at least equal to the minimum support threshold, or *minsup*, where

$$0 \leq minsup \leq 1.$$

The percentage of transactions that have the co-occurring items [27] of an itemset I is called a *support value* denoted by s, which can be calculated by:

$$s = \frac{N_{I^p}}{N} \tag{2.1}$$

, where N_I^p is the number of transactions that have I^P, N is the total number of transactions in the data set, and $0 \leq s \leq 1$.

The process to identify frequent itemsets is iterative. It starts from itemsets of size one, i.e. $I^1 \mid I^1 \in \{\{i_j\}, \{i_k\}, ..., \{i_m\}\}$. The process generates frequent itemsets of size l+1 from those of size l by using a database scan to gather the probability of each itemsets. However, it is learned that frequent itemsets possess a downward closure property, which states that all subsets of a frequent itemsets are frequent [1, 26, 66]. This property also implies that a super set of a non-frequent itemset cannot be frequent. For example, if I^1 is not frequent, then $I^2 = \{i_1, i_2\}$ cannot be frequent. The downward closure property helps prune the candidate set, which contains potential frequent itemsets of size l+1. The actual support values of the candidate itemsets will be verified by the subsequent database scan.

Therefore, for each iteration:

1.) the size of the frequent itemsets grows longer,

2.) the number of frequent itemsets of size *l* is less than or equal to those of size *l*-1, and

3.) the number of itemsets, which are the member of the candidate set of size *l*+1 is reduced.

The iterative process stops when at least one of the following conditions is met:

1. the process cannot determine a frequent itemset of size *l*+1, which has $s \geq minsup$, from the candidate set, or
2. $l+1 > M$.

After the above iterative process completes, the next step is to generate association rules (AR) from the frequent itemsets. The AR process uses a minimum *confidence* threshold, or *minconf*, where $0 \leq minconf \leq 1$, to determine whether a pattern (*antecedent* → *consequent*) is an association rule. A confidence value α of a pattern $I^x \rightarrow I^y$, where I^x is the antecedent, and I^y is the consequent of the pattern, can be calculated by

$$\alpha = \frac{s(I^x, I^y)}{s(I^x)} \tag{2.2}$$

, where $s(I^x, I^y)$ is a support value of the itemset that has I^x and I^y. In other words, α represents the conditional probability of I^y given I^x, or $P(I^y \mid I^x)$, where $I^x \cap I^y = \phi$, α determines the dependency of I^y on I^x [66].

It is worth mentioning that a frequent itemset of length *l* means that there are also 2^l - 2 subsets which are also frequent [4, 67] based on the downward closure property. These subsets of a frequent itemset increases the complexity of the algorithm [49]. Even

though, the idea of a candidate set helps improve Apriori's efficiency, but it is still time consuming to do database scans [51, 68]. Moreover, scans may also lead to memory exhaustion problems, especially when *l* is long [18, 49].

2.2 Statistical Analysis, Information Theory, and Other Measures for Association Mining

Statistical probability is a common background for both the information theory and data mining fields of study. This is simply because these fields share similar objectives in discovering *structure* in data [38]. We can date back the idea of the information theory to the early 1900s [69]. The well-known Shannon entropy [70], or in short "the entropy", is closely related to probability calculations and other statistical approaches. The entropy's purpose is to measure an information gain from knowing a piece of information [71]. A direct extension of the entropy in the information theory is Mutual Information Measure (MIM) [37], which calculates how many multiple pieces of information are related to one another.

Moreover, data mining research, such as [15, 30, 40-44, 57] suggested that the use of basic frequency counting and probability calculations of the support value [12, 13, 28] may not be sufficient in selecting information pieces (or items). This is because the support value may be able to represent co-occurrences of the information pieces, but it can merely justify how closely related those information pieces are. A statistical data mining related research [37] also agrees with this idea because the support value cannot provide a proof of whether the associations have or have not occurred by chance and holds any significant statistical meaning.

More research, such as [16, 23, 24], attempted to avoid using the support value as a threshold, with the reasoning that it is difficult to set a right value, and that the number of rules discovered can be extraordinary large if the support threshold is set too low. This problem is directly related to a tradeoff between the sensitivity and specificity [31, 72] of the support threshold. The sensitivity may be improved by lowering the threshold value, which may lead to uncovering more valuable associations that are often not trivial. On the other hand, the specificity may be improved by raising the threshold value, which may filter out unimportant associations that appear too rarely.

Extensions of DCM utilize the information theory through the correlation analysis as part of threshold values in selecting frequent itemsets. A purpose is to filter out trivial frequent itemsets that are not statistically correlated. However, since DCM partitions the data into domain-concept partitions, a partition size represents the actual distribution of the (*attribute*: *value*) domain-concept. Therefore, the *global* support value that is used among all of the domain-concept partitions is automatically and locally adjusted by the sizes of the partitions. As a result, the support value can be more sensitive and specific for each partition. Moreover, an association rule "domain-concept → itemset" is implied for every frequent itemset uncover from each domain-concept. Hence, there is no extra calculation needed. Further details can be found in Chapter 3.

2.3 Improvement of Frequent Itemset Mining

Frequent itemset mining is the most time consuming task of the association rule mining [51, 68]. *Frequent Pattern Tree* (FPT), proposed by Han et al. [20], which is used extensively in our research, utilizes an idea of a suffix tree (or a suffix pattern) [73] to

compact the data set. FPT and its variations [27, 46, 47] are considered as one of the most efficient frequent itemsets mining algorithms due to its data structure, limited number of the database scans (two scans to be exact), and the fact that the algorithm bypasses a need for candidate set generations . Li and Hamilton [48] also utilize a suffix tree to find pattern rules but with an aim to reduce itemset redundancy; and thus, reducing the number of rules discovered.

Lin and Kedem [4] proposed an algorithm called *Pincer-Search* to produce only a maximum frequent set (i.e. a set of the longest possible frequent itemsets in a top-down fashion). Pincer-Search is opposite to many association rule (AR) mining algorithms, which find frequent itemsets in a bottom-up fashion. Other maximum frequent set research includes [74, 75]. It is worth mentioning that a disadvantage of discovering only the longest possible frequent itemsets is that the association rules cannot be inferred from the longest frequent itemsets due to the loss of the subsets' support values [49].

Another alternative approach to frequent itemset is frequent "closed" itemsets [21, 49, 51, 52]. Generating frequent closed itemsets minimizes the frequent itemset redundancy while maintaining the ability to generate association rules. A frequent closed itemset is the super set of frequent itemsets. All of the transactions of the subsets of the frequent closed itemset are maintained. In other words, a frequent closed itemset is an itemset that passes the minimum support threshold, and there exists no supper set that completely contains both its items and its transactions. An algorithm, called "CHARM" (Closed Association Rule Mining, where the "H" is gratuitous), has been proposed to efficiently mine frequent closed itemsets [49, 50] . Please note that CHARM is one of the

approaches implemented by DCM for its offline mining processes (more details in Chapter 5).

2.4 Association Mining and Result Organizations

In general, after frequent itemsets and association rules have been mined, data mining processes have yet to be finished because there can be overwhelming amount of association rules than one could possibly utilize in a useful fashion [21, 22]. There are a number of investigators who focus on the organization, search, pruning, summarization, and access to data mining results [22-24, 48, 53]. For example, Dai and Huang [23] organized discovered association rules in a hierarchical fashion. Their organized rules are called General-Specific (GS) patterns. Another work is done by Gionis et al. [53] who used the randomization, Markov chain approach with clustering, and ranking as the basis of their approach to access data mining results. Please note that the Markov chain approach is closely related to the statistical probability and Bayes Theorem [76].

Some of the authors who recognized the need of reducing the number of rules after they have been discovered are such as Liu et al. [26]. The authors used Chi-square test to remove the insignificant associations, and used the direction setting (DS) rules as rule summaries. DS rules are groups of rules based on domain, where users can browse each DS rule to find more details.

The association organization offered by DCM is rather different from all of the afore-mentioned research. DCM organizes data before association mining by partitioning them into domain-concept partitions. More importantly, a domain-concept represents a characteristic of the data; hence, a domain-concept partition represents the distribution of

a certain characteristic. After the partitioning step, DCM analyses the partitions for frequent itemsets. Therefore, the frequent itemsets are automatically organized according to their domain-concepts. Further discussion can be found in Chapter 3.

2.5 Number of Transactions Reduction through Partitions and Random Sampling

It has been known that many applications apply data mining onto their very large data sets to find associations, patterns, or correlations, with a hope of being able to draw some conclusions or uncover valuable information from the data [1, 10]. We have discussed previously that the traditional AR mining may not be efficient enough to mine an entire data set at once due to its computational complexity [51, 68]. An attempt to solve the computational complexity problem is to partition the data in order to reduce the size of the data set per a data mining process. The discussion in this section will also be related to distributed or parallel processing because they usually go hand in hand with data partitioning.

Lee et al. [55] proposed the usage of a sliding-window to incrementally mine the data, partition by partition, with some overlapping between neighboring partitions. Schuster and Wolff [56] proposed an algorithm called *Distributed Association Rules Mining* (D-ARM) to mine associations from a data set that has been partitioned and physically placed across networks. Wei et al. [58] proposed a principle called *Data Feature Oriented Data Partitioning* (DFDP) to emphasize utilizing partitions in parallel computing for efficiency in data mining and for discovering more interesting rules. Another system, called *Association Rule Clustering System* (ARCS), uses clustering

technique (also widely used in artificial intelligence and machine learning fields [77]) to find associations from segment of data in two-dimensional space [19]. ARCS is considered efficient comparing to C4.5 decision tree [78] because the bit-wise operations that ARCS utilizes in association rule mining.

Unfortunately, many of the afore-mentioned data mining with partition research may potentially suffer from: 1.) communication overheads that need to be maintained among partitions to ensure that an itemset (which may or may not be frequent in some partitions) will be considered for subsequent comparisons [56], and 2.) an estimation or heuristic technique that may aid the maintaining of these itemsets from partition to partition [2, 55, 79].

A definite pruning (of the candidate frequent itemsets) technique commonly used by the association mining with partition is based on the downward closure property [1, 26, 53]. However, the property may be applied to association mining with partition only when an itemset is not frequent in any of the partitions by an inference that it will not be a frequent itemset. However, no conclusion can be drawn from other situations (i.e. whether an itemset that is frequent in one or more partitions will be frequent in the end) to further prune the candidate frequent itemsets. Therefore, an itemset uncovered from a partition with a support value that falls in a "border" range may be kept for the subsequent processes for further validations [2, 79].

One way to estimate the support value is to apply a statistical sampling technique to the entire data set [59]. The purpose is to reduce the number of itemsets that will be considered as a set of candidate frequent itemsets. The estimated support value will be updated when more partitions are mined in order to decide whether the itemset is in fact

frequent or not. Moreover, most data mining approaches that need to generate a set of candidate frequent itemsets require higher memory usage than the approaches that do not need a candidate set. Many data mining works, such as [40-45], have utilized statistical sampling technique beyond an estimation of the support value. The common idea between statistical sampling technique and association mining with partition is to speed up the mining process by reducing the number of transactions to be considered by attempting to maintain a level of accuracy of the results.

DCM utilizes the idea of data partitioning in order to reduce the computational complexity by applying a frequent itemset mining algorithm to each partition instead of to the entire data set. Since DCM partitions a data set with a main purpose to organize the data before analysis, a partition does contain all domain-concept qualified transactions. Therefore, DCM can mine these partitions independently, or in batches of distributed or parallel fashion. Moreover, a frequent itemset uncovered from a domain-concept partition is frequent with regard to the domain-concept. This means that DCM does not need a communication among processes or an estimation or border of the support value. More importantly, a DCM Partition Aggregation (DCM-PA) approach is developed in order to achieve a complete set of organized results by aggregating multiple domain-concept partitions efficiently without a mining process required. Hence, it takes less computation resources to find out whether a frequent itemset of a partition will stay frequent when its frequency is verified against the entire data set. Further details of DCM and DCM-PA are discussed in Chapters 3 and 4, respectively.

2.6 Incremental Data Mining

Most data sets have a nature of being collected or updated accumulatively, either on a schedule basis such as monthly, quarterly, or yearly [62], or in no exact temporal pattern such as the data that come in stream [63]. Many data mining researchers, including [64, 65, 79], have seen the need of mining new data and then combining the previous mining results to minimize the computational resources used. It is worth mentioning that the newly obtained data may update by either invalidating or strengthening the previously mined results [51, 64]. A proposed method, called "Temporal High Utility Itemsets – Mine" (THUI-Mine) [61], can efficiently mine data using a two-phase method along with borders [59] and with an extension of the sliding-window filter [55]. THUI-Mine's phase I is to *overestimate* itemsets, so it would less likely miss any frequent itemsets. Its phase II is to prune the overestimated itemsets with one database scan. The sliding-window filter is used so that the approach can incrementally process partitions of data. Therefore, this method can be adapted to be used for temporal data from stream.

A regression-based method, called Frequent Temporal Pattern of Data Streams (FPT-DS) [40] mines data based on a frequent pattern tree (FPT) approach [20]. The FPT approach itself is memory efficient because the FPT data structure is compact, yet can completely store all frequent itemsets. FPT-DS mines data by using an estimation technique to predict frequency of patterns of new data from experience of previously collected data. FPT-DS also uses regression analysis as its framework.

A need for real-time incremental data mining approaches has been studied. Thuraisingham et al. [60] suggested that in order to achieve real-time data mining, one

would need to combine various computational approaches. These include parallel computing, classification, clustering, link analysis, anomaly detection, and association rules. Combinations of them would quickly mine and detect patterns in on-going data. Moreover, this branch of data mining may also need efficient and sensitive input devices, such as surveillance cameras, to reliably feed a quality incoming data stream quickly. This is because the validity of mined results directly depends on the quality of the input. However, data that comes with this fast pace, oftentimes contain noises, or the data may be incomplete or inaccurate at the certain time point. In conclusion, real time data mining is still an on-going research topic that is receiving much attention from researchers.

DCM can apply its domain-concept partitioning technique to mine data that are in incremental or temporal nature by using the schedules of the data as the domain-concept partitions. Further, results from the domain-concept partitions can be used as temporal patterns to build trends for result comparisons. The functionality to build trends is offered by a DCM Web system, called "DCMiner" (more details in Chapter 5).

For the issue of updating the current results with the newly mined results, a DCM Partition Aggregation (DCM-PA) approach can be utilized. DCM-PA applies Bayes Theorem [31] and the pipelining technique [34] to efficiently aggregate results from domain-concept partitions, where the two major partitions are the previously collected and the newly collected data sets. Moreover, each of the two major partitions is composed of sub-partitions, which are regular (*attribute*: *value*) domain-concepts. Further details regarding DCM-PA can be found in Chapter 4.

CHAPTER 3

DOMAIN-CONCEPT MINING (DCM)

3.1 Domain-Concept Mining Approach

Domain-Concept Mining (DCM) approach is developed to address the following problems in a descriptive association mining research:

1.) association mining efficiency and memory exhaustion,

2.) organization and management of the overwhelming amounts of association mining results,

3.) inability to discover valuable (*attribute*: *value*) pairs of under-represented groups of population, where these pairs often have low support values, and

4.) a problem of an insensitive and unspecific global minimum support threshold.

The processes of DCM, which is depicted in a DCM flowchart (Figure 3.1), include:

1.) Input data pre-processing,

2.) Data partitioning, which includes a statistical power analysis for a sample size estimation,

3.) DCM offline association mining, and

4.) DCM Partition Aggregation (DCM-PA), which will be detailed in Chapter 4.

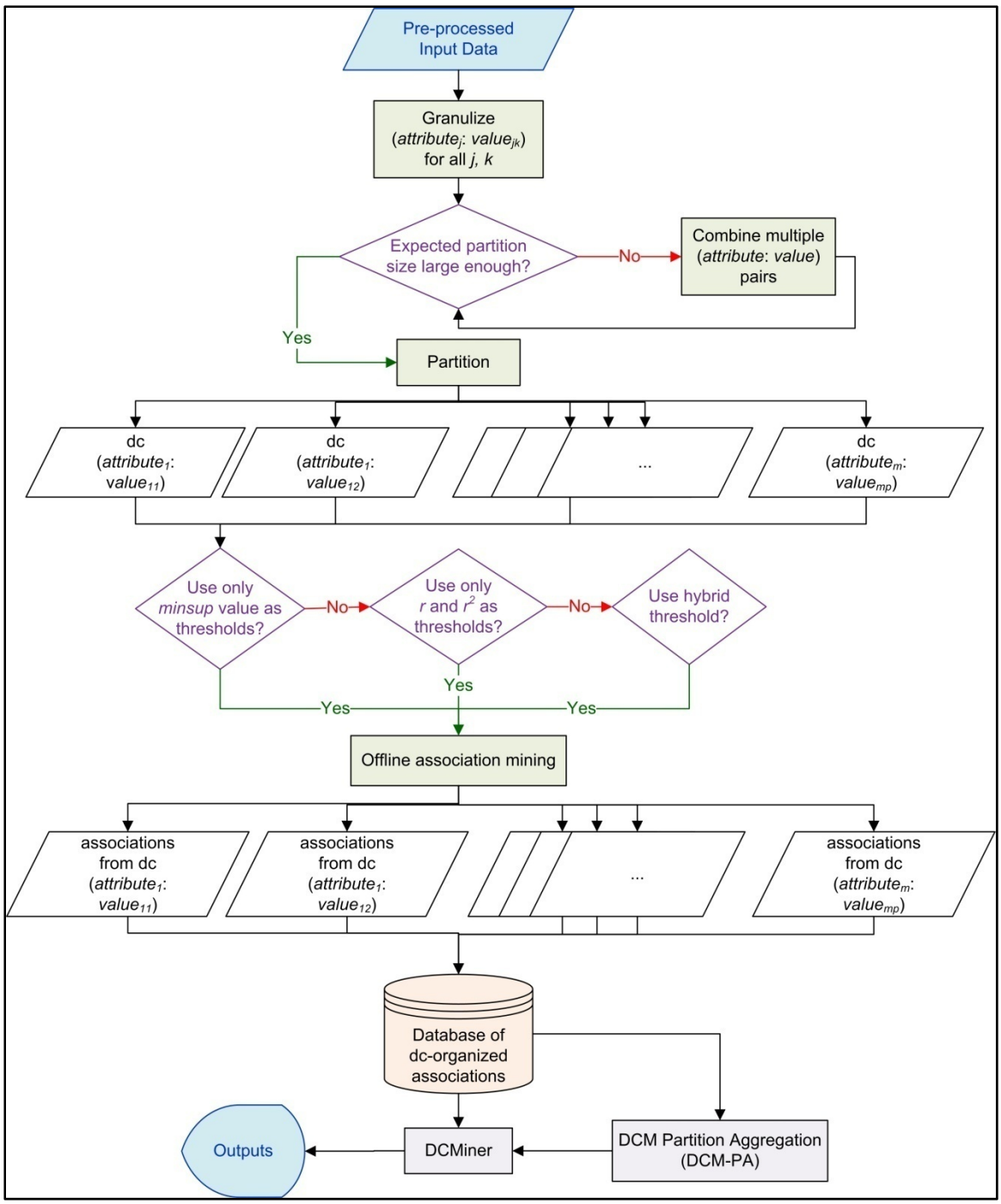


Figure 3.1. A Domain-Concept Mining flowchart.

3.1.1 DCM Pre-Processing Step

The initial step of DCM is data pre-processing. It starts by identifying domain-concepts (dc). In general, a dc is a set of (*attribute*: *value*) pairs, or *items* as indicated by the user who is a subject-matter expert. However, in order to systematize the process and maximize the independency between the system and the human expert involvement, DCM can granulizes all dc's to contain only one item by pairing each $attribute_j$ to each of its $values_{jk}$'s. This step assumes that attributes' values are categorical. However, if the values are originally continuous, DCM discretizes them by utilizing pre-established scales, such as age groups defined by the "Center for Disease Control and Prevention" (CDC). In a circumstance that the scale is not applicable or unavailable, DCM applies a statistical analysis to bin a continuous value around their average value based on an assumption of the normal distribution and the central limit theorem [31].

After the granulizing is done, each of the ($attribute_j$: $value_{jk}$) pairs is also uniquely identified as an item using an enumerated value, called "itemID", as its representative. An example of DCM pre-processed biomedical informatics attributes (which is a small portion of the BRFSS 2006 data set [33]) are shown in Table 3.1.

Table 3.1. An Example of DCM Pre-Processed Biomedical Informatics Attributes

ItemID	Name	Description	Value	Meaning	Interesting Indicator
1	*GENHLTH*	Would you say that in general your health is:	1	Excellent	N
2	*GENHLTH*	Would you say that in general your health is:	2	Very good	N
3	*GENHLTH*	Would you say that in general your health is:	3	Good	N
4	*GENHLTH*	Would you say that in general your health is:	4	Fair	Y
5	*GENHLTH*	Would you say that in general your health is:	5	Poor	Y

ItemID	Name	Description	Value	Meaning	Interesting Indicator
6	*HLTHPLAN*	Do you have any kind of health care coverage, including health insurance, prepaid plans such as HMOs, or government plans such as Medicare?	1	Yes	N
7	*HLTHPLAN*	Do you have any kind of health care coverage, including health insurance, prepaid plans such as HMOs, or government plans such as Medicare?	2	No	Y
8	*MEDCOST*	Was there a time in the past 12 months when you needed to see a doctor but could not because of cost?	1	Yes	N
9	*MEDCOST*	Was there a time in the past 12 months when you needed to see a doctor but could not because of cost?	2	No	Y
10	*DIABETE2*	Have you ever been told by a doctor that you have diabetes?	1	Yes	Y
11	*DIABETE2*	Have you ever been told by a doctor that you have diabetes?	3	No	N

Further, the pre-processing step also allows the user to identify what value(s) of an attribute that he or she may want to omit from the mining results. This identification is called "interesting indicator" with values Y or N. The indicators are considered beneficial because they may reduce the number of *trivial* items in the association results. Without these indicators, the trivial items are likely to be repeating as associations due to the fact that they are commonly populated in the data. Moreover, the DCM offline mining will be able to achieve a higher efficiency when there is less number of items to be considered per partition. On the contrary, the interesting indicators will not prevent the trivial items from being used as dc's. Having a complete set of items as dc's (regardless whether the items are trivial, non-trivial, or from the under-represented groups of the population) enables:

1.) Human experts to compare and contrast the association results, especially for the findings that are different from their assumptions or prior knowledge. The experts can also validate findings of a dc against other dc's,

2.) Trends to be formed among different dc's. These trends reflect changes and transitions of associations from one dc to the next.

3.) More flexibility and choices that DCM-PA could offer to the experts during the on-demand partition aggregation step.

4.) DCM-PA to aggregate a set of dc's that share the same $attribute_j$ in order to obtain a complete set of associations. This is because aggregating all values of an attribute is the same as combining all transactions of the entire data set together.

5.) DCM-PA to fully utilize Bayes Theorem to infer values of aggregation results "on-the-fly" without accessing the original un-mined data. More details of the on-demand partition aggregation and how DCM-PA aggregates multiple dc partitions will be discussed in Chapter 4.

3.1.2 DCM Partitioning Step

As discussed previously, the most granulized partitioning criterion is obtained by a set of one (*attribute*: *value*) pair. Each of the pairs filters qualified transactions for a dc partition. Let x be an item, which is an (*attribute*: *value*) pair in a database D, X be x's domain-concept, T be the set of all transactions in the database, *itemID* be the selected items that theirs interesting indicators (or *ini*) values are 'Y' and exclude the *itemID* of the criterion x, ID(.) be a function that transforms a criterion to an *itemID*, n be the total number of items, and π and σ be the relational algebra projection and selection operators [34], respectively. A subset of T that shares the same domain-concept X is defined as:

$$T_x = \pi_{itemID \in \forall_{i=1}^{n} \mathrm{ID}(\{\zeta_i \setminus x\})}(\sigma_{x,(ini='Y')}(T)) \tag{3.1}$$

With the DCM partitioning technique, one transaction could be in multiple partitions because the transaction may qualify for many domain-concepts according to its attribute values. The set of transactions will then be mined by the DCM offline association mining process to extract frequent itemsets. An itemset *i* is said to be "frequent" in a domain-concept *X* if and only if its support value (*s*) is greater than or equal to the minimum support threshold value (*minsup*). The support value of an item *i* can be calculated by:

$$s = \frac{|T_X^i|}{|T_X|} \tag{3.2}$$

, where T_X^i is the set of transactions in *X* that contain *i* and |.| indicates the number of transactions of a data set. It is worth mentioning that the *minsup* criterion is a "global" value used for all of the domain-concept partitions. More importantly, the value of the *minsup* is automatically and proportionally adjusted by the size of a domain-concept. For example, a global *minsup* of 0.1 (or 10%) when used by a domain-concept partition with 10,000 transactions yields a different minimum transactions from the same *minsup* of another domain-concept partition with 20,000 transactions. Also important is the fact that $\frac{|T_X^i|}{|T_X|}$ is equivalent to a conditional probability of P(*i*| *X*), which is the confidence value (α) of an association rule "$X \rightarrow i$".

To demonstrate the partitioning process, the attributes (shown previously in Table 3.1) and an example of transactions of a data set (shown in Table 3.2) are used. According to Table 3.1, there are 11 dc's identified. Let Table 3.1 be called "items", Table 3.2 be called "data", and both of them are stored in a database system. DCM

partitions “data” by using a Structured Query Language (SQL) statement [34]. Figure 3.2 is an example of a statement used when partitions the data for (*DIABETE2*: *1*) dc partition.

Table 3.2. An Example of a DCM Pre-Processed Data Set

TID	itemID
1	1,6,9,11
2	2,9,11
3	3, 11
4	4,7,8,10
5	5,7,8,10
. . .	. . .
m	5,6,9

```
SELECT *
FROM     data, items
WHERE (items.Name = 'DIABETE2') AND
          (items.Value = 1) AND
          (items.itemID IN data.itemID);
```

Figure 3.2. A SELECT statement used by DCM to partition data.

However, to better utilize the human experts’ interesting indicators of the items to compact the size of the dc partitions, the results from the previous SELECT statement can be further reduced. The reduction can be achieved by: 1.) keeping the previous results in a temporary table, called “temp,” and 2.) implementing the SELECT statement as shown in Figure 3.3.

```
SELECT temp.itemID
FROM   temp, items
WHERE  (items.InterestingIndicator = 'Y') AND
       (items.itemID = temp.itemID);
```

Figure 3.3. A SELECT statement used by DCM to filter (attribute: value) pairs' itemIDs with interesting indicator = 'Y'.

In conclusion, the three major benefits from this DCM partitioning step are:

1.) DCM organizes the data into dc partitions; hence, the data mining results will be organized accordingly,

2.) DCM needs only one *minsup* threshold to be used by all dc partitions. However, this threshold is automatically adjusted by the total number of transactions of a dc partition,

3.) All dc partitions are independent and compact because:

a.) A dc partition contains *all* transactions that share the same dc,

b.) Only itemIDs with interesting indicator 'Y' are included.

3.2 DCM Offline Mining

After the initial partitioning step, DCM implements a well-known and efficient brute-force frequent itemsets mining algorithms called *Frequent Pattern Tree* (FPT) [20]. This is done in order to conduct DCM offline mining, at the rate of one FPT process per partition in batches of distributed or parallel fashion. To continue the case examples used

in the previous sections, there are 11 dc partitions to be processed. Furthermore, it is worth mentioning that a mining process can be independently executed with no communication overhead among the dc partitions.

Following the offline mining step, DCM stores all of the results that are associated with their domain-concepts in a relational database. The relational schema of the offline mining results and the entity-relationship diagram (ERD) are shown in Figure 3.4 and Figure 3.5, respectively.

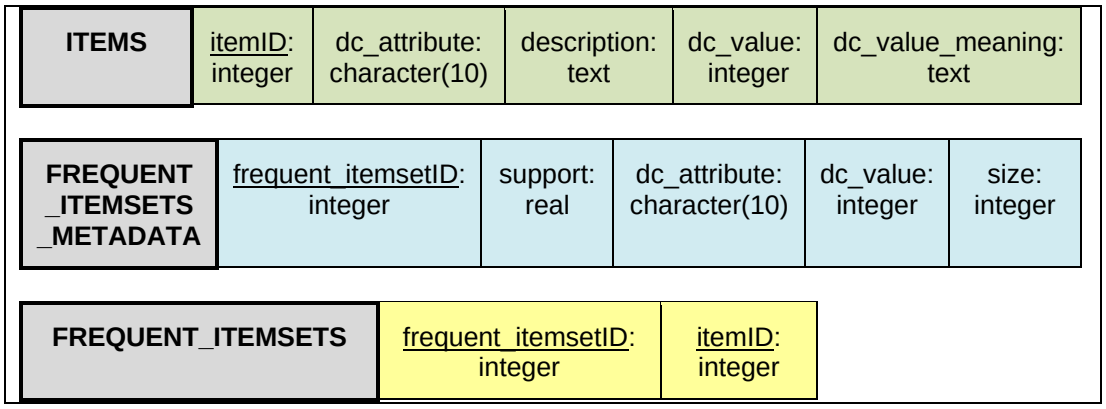


Figure 3.4. The relational schema of the frequent itemsets from DCM offline mining processes.

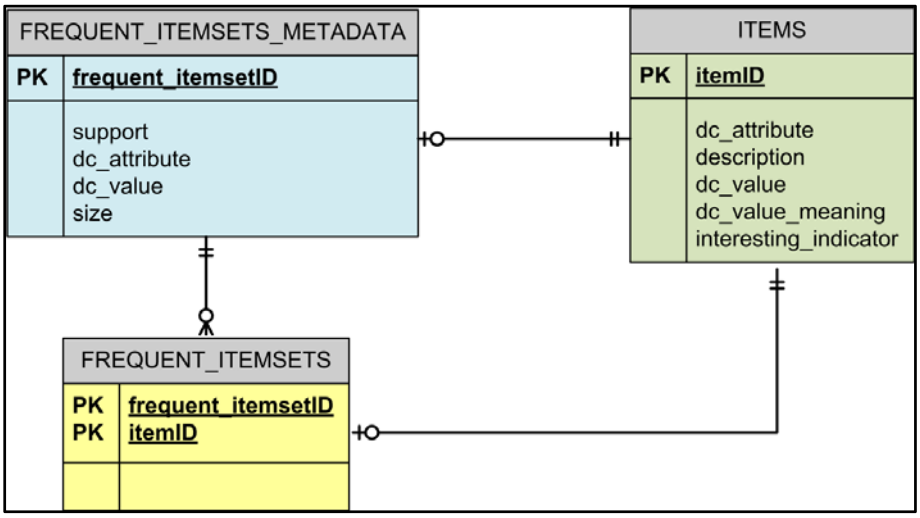


Figure 3.5. The entity-relationship diagram for the relational schema as shown in Figure 3.4.

There are three relational tables that are related to the DCM offline results. The first table is “ITEMS”, which also resembles to Table 3.1. This table contains: 1.) itemID, which is its transaction unique identifier or primary key, 2.) dc_attribute, 3.) description, which describes the dc_attributes in full text, 4.) dc_value, 5.) dc_value_meaning , and 6.) interesting_indicator. The second table is “FREQUENT_ITEMSETS_METADATA”. This table contains: 1.) frequent_itemsetID, which is the primary key, 2.) support, 3.) dc_attribute, 4.) dc_value, and 5.) size, which is the size of an itemset (how many co-occurring items in an itemset). As shown in Figure 3.5, there is a relationship between FREQUENT_ITEMSETS_METADATA and ITEMS. FREQEUNT_ITEMSETS_MATA DATA’s dc_attribute and dc_value reference to ITEMS’s dc_attribute and dc_value. The third table is “FREQUENT_ITEMSETS”. It contains two attributes, frequent_itemsetID and itemID, where the primary key is a composite key (meaning multiple attributes are needed to form a unique identifier). This is because frequent_itemsetID alone cannot be a unique identifier because an itemset can contain one or more items. FREQUENT_ITEMSETS’s frequent_itemsetID references to FREQUENT_ITEMSET_ METADATA’s frequent_itemsetID, and FREQEUNT_ITEMSETS’s itemID references to ITEMS’ itemID.

Furthermore, these relational tables are utilized by: 1.) the on-demand DCM Partition Aggregation (DCM-PA) system, and 2.) the online associations viewing system, called “DCMiner”. DCMiner is an interface for the users to utilize the results. It is a tool to browse, compare, contrast, aggregate, and view the results using various visualization techniques. More of DCM-PA’s and DCMiner’s details will be discussed in Chapters 4 and 6, respectively.

Please note that for an exploration of an efficient algorithm used during the offline step, DCM also utilizes *CHARM* (or "Closed Association Rule Mining", where the "H" is gratuitous) [49] as an alternative to FPT. Specifically, CHARM offers a more compact result set, i.e. a set of frequent "closed" itemsets, as previously discussed in Chapter 2. The relational schema and ERD of the frequent closed itemsets are the same as those of the FPT's frequent itemsets. Further experimental results and comparisons between FPT and CHARM that are implemented by DCM are discussed Chapter 5.

In addition to DCMiner, a main contribution of DCM is its partition aggregation approach, called "DCM-PA", which can aggregate results among dc partitions on-demand and online based on Bayes Theorem [76, 80]. DCM-PA offers two ways to aggregate the results: 1.) by broadening (union), and 2.) by narrowing (intersection) the dc partitions. The aggregation can be done efficiently because: 1.) DCM-PA reuses the information obtained from the offline mining step, which are stored on a relational database, and not in the main memory, 2.) the calculation of aggregated results' support values are done by inferring and propagating the support value of the offline results, and 3.) there is no computationally expensive association mining process involved.

Specifically in regards to the union ability of DCM-PA, DCM can mine the data offline through the steps we have discussed in this Chapter, and then DCM-PA aggregates the results from sets of dc partitions that share the same attribute. A purpose is to achieve a complete set of results with respect to a dc attribute. A set of dc partitions is such as (*age group*: *), where * represents all age groups' values. For example, DCM-PA combines (unions) all associations of the age group dc partitions together. This way, the

results of this union operation is equivalent to the complete set of the associations as if the entire data set was mined from the ground-up.

Together, the DCM and DCM-PA approaches can also be extended to mine: 1.) temporal data sets (e.g. mine each temporal unit, such as month, quarter, or year as a dc partition, and then aggregate the partitions), 2.) spatial data sets (e.g. mine each specific data region such as county or state as a dc partition, and then aggregate them), and 3.) incremental data sets (e.g. mine the newly collected data as a dc, and then combine the results with the previously mined ones). When DCM-PA aggregates partition of these dc's, the associations are updated. For example, the new associations from an incremental dc partition can change the support values of the historical associations. Further details of DCM-PA are in Chapter 4.

3.3 An Extension of DCM with Statistical Analyses

In this section, we discuss plans of theoretically extending DCM by incorporating statistical analyses in: 1.) strengthening the threshold value for determining associations by using correlation values, and 2.) determining the domain-concept size by applying statistical power of estimation analyses.

3.3.1 Correlation Analysis

Data mining is a multi-disciplinary computational approach with probability and statistical analyses as its major foundations. There have been many implementations of such analyses in data mining, ranging from very basic frequency and probabilistic approaches (used in determining support and confidence values of the traditional AR

mining) to correlations, regression, multivariate, and principle component analyses, just to name a few [30]. The main goal of data mining is to find associations, correlations, or patterns in the data that may be non-trivial. The data mining findings are generally called *discovered knowledge*, where the purpose of discovering the knowledge is to increase the utilization of the data through a summarization offered by the knowledge.

To only implement the frequency, which calculates probability (or *proportion*) value, seems insufficient to serve the aforementioned purpose. This is because the knowledge found using these values is not always relevant and may sometimes happen by chance [39]. Brin et al. [15] have initiated the use of a statistical technique, the chi-square test, which may be considered as an added measurement to the probability and conditional probability used widely in the traditional AR mining. The chi-square test has the ability to measure the significance of itemsets by forming a border between absence and presence of correlations. The itemsets are crucial to the subsequent process of forming association rules. This is simply because if the frequent itemsets (the itemsets that pass the minimum support threshold) are not properly discovered, then it is rather less likely to achieve a valid set of association rules afterwards.

The approach proposed by Brin et al. [15] strengthens association mining by considering both positive and negative correlations in filtering itemsets. An example of a classic market basket analysis (Table 3.3), which can show why the support and confidence values may not be sufficient in judging what itemsets are to be used to build association rules, is as shown in Table 3.3.

Table 3.3. An Example of Market Basket Transactions between Two Items, Beer and Diapers, in Summary

Items	Diapers	No Diapers	**Total Rows**
Beer	20	10	30
No Beer	55	15	70
Total Columns	75	25	100

Assume that we set a minimum support threshold (*minsup*) to be 0.2 (or 20%) and a minimum confidence threshold (*minconf*) of 0.6 (or 60%). The support value of people who buy beer and diapers in the same transactions is 0.2 (20/100). The confidence value of people who buy beer and also buy diapers (or {beer → diapers}) is 0.67 (0.2/0.3). The confidence value of people who buy diapers then buy beer (or {diapers → beer}) is 0.27 (0.2/0.75). Therefore, we can conclude that we found a frequent itemset of {beer, diapers} using the *minsup* = 0.2, and found an association rule of {beer →diapers} using the *minconf* = 0.6. It is worth mentioning that we did not find an association rule of {diapers → beer} using the same *minconf* threshold.

On the other hand, we want to analyze further whether beer and diapers really have a strong relationship. The appropriate statistical technique that we will use is the correlation analysis, called "Pearson correlation coefficient" (*r*) [31], between two random variables. In this case, the variables are beer and diapers. *r* can be calculated by:

$$r = \frac{n\ (\sum xy) - (\sum x)(\sum y)}{\sqrt{[n(\sum x^2) - (\sum x)^2][n(\sum y^2) - (\sum y)^2]}} \tag{3.3}$$

, where beer is represented by $x \in \{0,1\}$, diapers is represented by $y \in \{0,1\}$, an absence of an item is represented by 0, and a presence of an item is represented by 1.

Please note that the approach introduced in [15] used a rather intuitive calculation to determine a correlation (e.g. P({beer, diapers}) / [P({beer}) * P({diapers})] = 0.2/(0.3*0.75) = 0.89), which gives an alternative value to measure the dependency between beer and diapers. This is based on an assumption of independent events [31] that if two variables (or events) are independent then P({beer, diapers}) would result in the same value as P({beer})*P({diapers}); hence, the calculated dependency value is 1. Furthermore, when the dependency value is less than 1, it indicates a negative relationship between variables, i.e. the variables co-occur together in the same transactions less often than when they separately occur, and vice versa. The same calculation may also be implied as a measure of the dependency strength (i.e. variables are less dependent on one another when the value is close to 1).

However, DCM follows the Pearson correlation coefficient calculation in determining dependencies. As a result, the correlation coefficient *r* of the itemset {beer, diapers} calculated by equation (3.3) is -0.13, and the coefficient of determination r^2 is 0.02. The *r* value indicates that beer and diapers have a negative correlation, which is the same conclusion as [15], but in different scales. Furthermore, the r^2 value indicates that beer and diapers do not have a strong correlation, where $0 \leq r^2 \leq 1$ (the higher the r^2, the stronger the correlation). Thus, this demonstrates that the traditional association mining, which uses only the support and confidence thresholds, may not be sufficient in determining whether associations found are statistically correlated.

To analyze this insufficiency of support and confidence thresholds further, we can examine the absence of items (as denoted by ¬) of beer and diapers in the itemset {¬beer, ¬diapers}. Following the same settings of *minsup* and *minconf*, the support value of

{¬beer, ¬diapers}, and the confidence values of {¬beer → ¬diapers} and {¬diapers → ¬beer} are 0.15, 0.21, and 0.6, respectively. These numbers indicate that the absence of the items cannot pass the threshold values; hence, the itemset {¬beer, ¬diapers} is not frequent and cannot be considered as an association rule. More importantly, the correlation coefficient r and coefficient of determination r^2 values are exactly the same for both the original case and in case where the items are absent. (Please note that when we calculated the values with the absence of the items, we used 1 to represent an absence and 0 to represent a presence of an item).

Therefore, we conclude that the correlation coefficient r and coefficient of determination r^2 add more information regarding the strength and direction of the relationships of itemsets to the support and confidence thresholds, without being impacted by how frequent the items appear in the transactions. In addition, the r and r^2 calculations focus on the scope of the relationship between variables, not on how frequent these variables occur in the whole data set.

3.3.2 A Hybrid Threshold

From the prior discussion, it has been demonstrated that using only the primitive probability calculations of the support and confidence thresholds in the traditional AR approach may not be sufficient in describing relationships between variables. In addition, using only correlations may also not be a proper representation of how important (frequent) the items are to the whole data set.

In this section, the combination of both approaches is explained. We discussed a *hybrid threshold* (*h*), which is based on a weighted harmonic mean among the support,

correlation, and coefficient of determination values. The confidence value is omitted from the hybrid threshold due to the fact that the value is implied by the support value as previously discussed in Section 3.1.2.

To understand why the harmonic mean is chosen for the hybrid threshold calculation, let us begin with a comparison of mean values and calculations. There are three main types of mean calculation – 1) arithmetic mean, 2) geometric mean, and 3) harmonic mean, each of which is calculated as shown in Table 3.4.

Table 3.4. Formula of Arithmetic, Geometric, and Harmonic Means

Types of Mean	Simplest Formula	Most Informative Formula
Arithmetic mean of *x* and *y*	$(x+y)/2$	$0.5x+0.5y$
Geometric mean of *x* and *y*	$\sqrt{xy}$	$x^{0.5}\cdot y^{0.5}$
Harmonic mean of *x* and *y*	$2xy/(x+y)$	$\frac{1}{(0.5/x)+(0.5/y)}$

Moreover, Table 3.5 illustrates that there is no difference in how x and $(100 - x)$ can affect the values of the arithmetic means (the regular average values) as long as the summations of x and $(100 - x)$ stay the same. The same table also shows that the geometric and harmonic means can give penalties to inequalities between the two numbers, regardless of the same summation values. Please note that the examples shown below are adapted from [81].

Table 3.5. An Example of Arithmetic, Geometric, and Harmonic Means

x	100-*x*	Arithmetic mean	Geometric mean	Harmonic mean
50	50	50	50	50
40	60	50	49	48
30	70	50	46	42
20	80	50	40	32

Furthermore, Figure 3.6 shows the comparison between the geometric and harmonic means. One can see that the geometric means give fewer penalties to uneven numbers than the harmonic means. If each number in the mean calculation is considered as a measurement of good performance, and a better mean calculation should be able to express the evenly good performances, it follows that using the harmonic mean would be a better choice than the geometric mean.

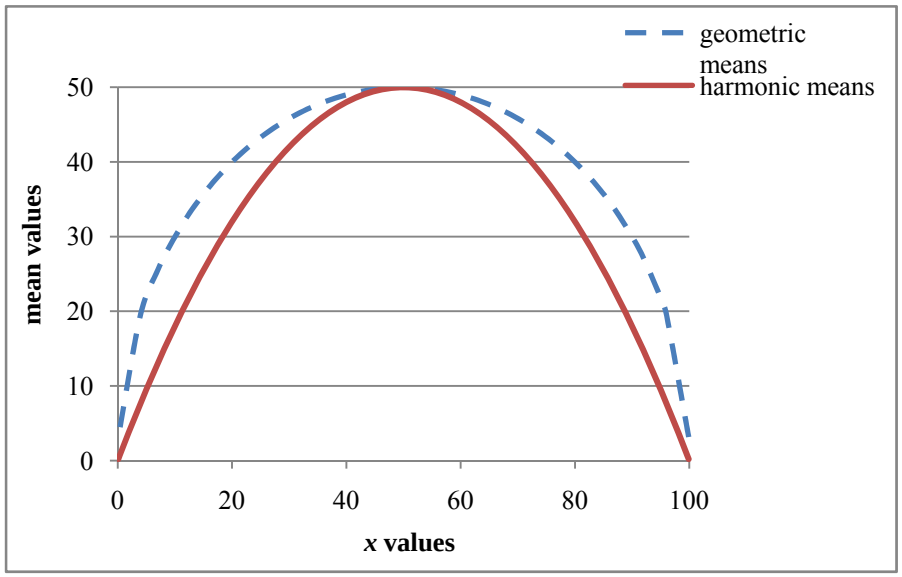


Figure 3.6. Geometric and harmonic means between two values: *x* and (100 - *x*).

An example of a mean calculation that utilizes the harmonic mean is the "F-measure" [71], shown in equation (3.4). F-measure is widely used in various information retrieval approaches. There is an important add-on to F-measure, which makes the calculation a *weighted* harmonic mean between the *precision* and *recall* values. The weight ω is to adjust how important the recall is for the F-measure value.

$$F_{\omega} = \frac{(1+\omega)\cdot precision \cdot recall}{(\omega \cdot precision) + recall} \tag{3.4}$$

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{3.5}$$

One can find from equation (3.4) that when the ω value increases, the weight of recall is higher. Equation (3.5) shows an example of evenly weighted precision and recall when ω = 1. To elaborate further with commonly used F-measures, F_2 weights recall twice as much as precision. On the contrary; $F_{0.5}$ weights precision twice as much as recall.

From the above discussions, the hybrid threshold (h_{ω}), which is a weighted harmonic mean of support (s), correlation coefficient (r), and coefficient of determination (r^2), can be calculated below.

$$h_{\omega} = \frac{(1+\omega)\cdot s \cdot r \cdot r^2}{(\omega \cdot s) + (r + r^2)} \tag{3.6}$$

The principle is to utilize previously explained threshold parameters – support, correlation coefficient, and coefficient of determination together as one value. The support of a frequent itemset of a dc partition is the confidence of (dc partition → frequent itemset) with respect to the entire data. More importantly, the support value is

given a specific weight ω, where the other values share an equal weight. It is worth mentioning that, based on the objective to discover non-trivial knowledge, a weight should be given less to the support value by setting $\omega > 1$.

3.3.3 Domain-Concept Partition Size Estimation

The DCM approach is designed to granulize the partitioning criteria by pairing each attribute with each of its values. Even though, the assigned domain-concepts are highly reliable because: 1.) all of their qualified transactions are grouped together, 2.) dc partitions are completely data driven, and 3.) there is no manipulation to the data and its distribution in anyway, we could foresee a few potential problems associated with this partitioning approach. Examples include the issue of under-represented groups of population with (*attribute*: *value*) pairs that occur too few in a data; and therefore, the findings from these domain-concepts may be less statistically significant.

Furthermore, an important development that compliments DCM is its partition aggregation approach, called DCM-PA (which will be discussed in details in Chapter 4), has changed the scenario. By being able to aggregate partition using the union operation (i.e. merge the transactions from partitions) may improve statistical values of the findings. However, it is still more suitable to prevent partitions from being too small so that all findings from any partitions can be presented to the users directly as statistically worthy findings.

Therefore, to prevent dc partition from being too small, we developed a systematic way of determining a domain-concept size based on a statistical approach of sample size estimation or power analysis [31, 82, 83], as a guideline to determine an

expected size of the domain-concept' lower bound value. The foundations of this approach are discussed as follows.

The basic idea of the sample size estimation value, *n*, is to find a statistically good interval ($\left[\left(\hat{p} - z_{\alpha/2}\sqrt{\hat{p}(1-\hat{p})/n}\right),\left(\hat{p} + z_{\alpha/2}\sqrt{\hat{p}(1-\hat{p})/n}\right)\right]$, see also Figure 3.7) that a value of interest ($\hat{p}$) can possibly fall in the confidence interval. However, this is also not to allow the interval to be too wide until it is too useless to estimate any values correctly (or not to allow the interval to contain too many transactions to be efficiently mined by DCM).

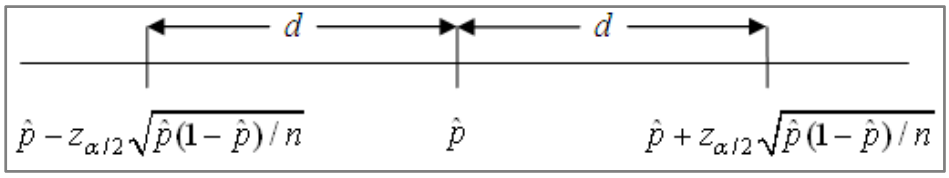


Figure 3.7. The confidence interval of $\hat{p}$.

In other words, the goal is to be 100(1-α) % certain that *p* (or a value DCM is estimating) falls in the interval of ($\hat{p}$ - *d*) $\leq \hat{p} \leq$ ($\hat{p}$ + *d*). A distance *d* is defined as:

$$d = z_{\alpha/2}\sqrt{\hat{p}(1-\hat{p})/n} \tag{3.7}$$

Furthermore, $z_{\alpha/2}$ represents a percentile of the unit normal curve (*z*) for the significance criterion of two-tailed test ($\alpha/2$). The value $z_{\alpha/2}$ can be looked up from the table of the standard normal distribution. One can solve equation (3.7) above to find the value of the sample size *n* as follow.

$$n \doteq \frac{z^2_{\alpha/2}\hat{p}(1-\hat{p})}{d^2} \tag{3.8}$$

It is also important to mention that $\hat{p}(1-\hat{p})$ is equivalent to σ, where σ is a standard deviation value. Further, to simplify equation (3.8), one can utilize the knowledge of calculus to conclude that the following equation gives the upper bound value of 1/4.

$$\text{MAX}(g(\hat{p})) = \hat{p}(1-\hat{p}) = 1/4 \tag{3.9}$$

Therefore, we can get a cleaner version of equation (3.8) below:

$$n \doteq \frac{z^2_{\alpha/2}}{4d^2} \tag{3.10}$$

, which can be used when no prior estimate is available. Please note that the above sample size estimation is designed for continuous variables. However, most of the attributes that DCM regularly faces with are not continuous, but categorical. Importantly, when DCM faces with a continuous variable (attribute), such as age, it discretizes the values into categories, e.g. age groups.

In general, the larger the sample size *n*, the smaller the error from a statistical analysis would be. Further, DCM partitions data with objectives to achieve a better efficiency (i.e. processing a smaller set of transactions would require less computational time and resources). Therefore, the ability to determine an appropriate partition size may improve the accuracy (of a correlation coefficient, for example) when DCM faces with domain-concept partitions that may have too few transactions.

A basic statistical concept of Type I and Type II errors (shown in Table 3.6) is reviewed to provide basis understanding of the sample size estimation. These errors are to be considered when one conducts a statistical testing whether to accept or reject a null hypothesis (H_0). A power of a statistical test can be obtained by using a complement of β, or (1- β), where "β is the probability of falsely accepting H_0 when in fact H_1 is true" [84].

Table 3.6. Type I and II Errors

		Hypotheses	
		H_0	H_1
Decisions	H_0	Correct acceptance of H_0	Type II error β
	H_1	Type I error α	Correct rejection of H_0

The goal is to minimize both of the error types. However, there is a trade-off between minimizing these two errors, e.g. if the error of Type I is lower, then Type II is higher.

The next step is to derive an *effect size* (d) which determines the minimum acceptable difference for a statistical test. A relationship between d and r proposed by *Cohen's* standardized difference, d, [85], as follows.

$$r = \frac{d}{\sqrt{(d^2 + 4)}} \tag{3.11}$$

One can rearrange equation (3.11) to compute d from r as follows.

$$d = \frac{2r}{\sqrt{(1 - r^2)}} \tag{3.12}$$

, where r is an expected value of the correlation coefficient from a domain-concept. In equations (3.11) and (3.12), we consider only an absolute value of r by ignoring the correlation directions (+/-).

The standardized difference or effect size, d, is basically a scale of magnitudes for changes in means [83, 86]. DCM sample size estimation that is based on a correlation value r will allow some acceptable changes in the mean of an attribute that is being considered as the domain-concept. This also enables DCM to tolerate some *noise* in data, which usually occurs in a real-world situation.

The following is a sample size estimation through a calculation that brings in an effect size (d) and a correlation r together, which is introduced in [83, 86]. The calculation takes into account a confident interval of 95%, which implies Type I error (α) to be 5%, and an acceptable value of Type II error (β) of 20%.

$$n = \frac{32}{d^2} \tag{3.13}$$

By designating Type II error to be 20% implies a conservative estimate of a power of a test to be only 80%. Furthermore, an approach for a sample size estimation, such as the calculation shown in equation (3.12), that is related to a ratio between α and β is also called *Compromise Power Analysis* based on *A Priori Power Analysis* [84]. A Priori Power Analysis is an analysis that involves samples without a prior estimate. The power of a test, 1- β , for 800 samples, can be shown in Figure 3.8. The figure depicts two normal distribution curves, which are plotted by using a statistical power analysis tool, called *G*Power* version 3.0.10 [84].

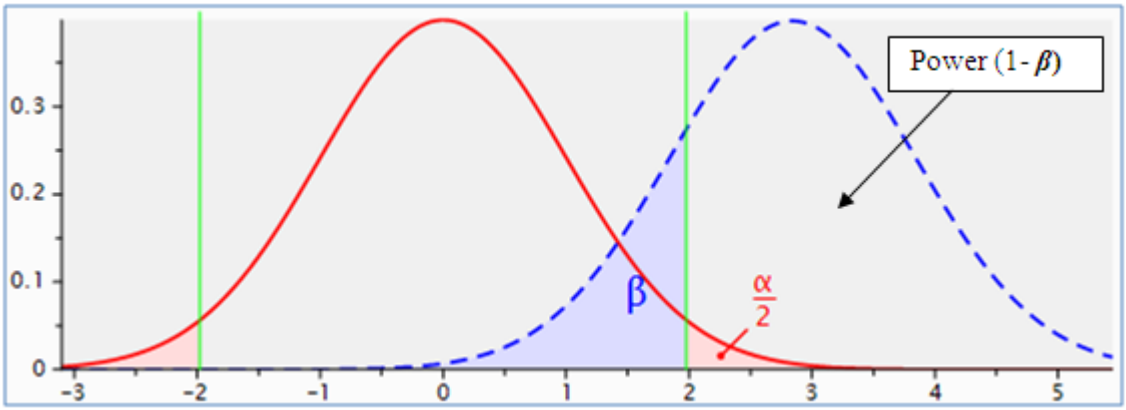


Figure 3.8. A Power analysis of sample size 800 with a ratio between α and β of ¼.

To further simply the sample size estimation, DCM pre-calculates the estimated d values using equation (3.12) and the corresponding sample sizes (n) using equation (3.13). This is to generate a look-up table shown in Table 3.7. Please note that the rough scales of r, which includes "trivial", "small", "moderate", "large", "very large", and "nearly perfect" is based on a suggested scale of Cohen [87]. The scales are for a purpose of brief interpretations.

Table 3.7. A Lookup Table for Effect Size (d)

Correlation Scales	Trivial		Small		Moderate	Large		Very large		Nearly perfect	
$\|r\|$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
d	0	0.2	0.4	0.6	0.9	1.2	1.5	2.0	2.7	4.1	∞
n	n/a	800	200	89	40	23	15	8	5	2	n/a

For example, when d = 0.2 (or we expect a small correlation between variables), equation (3.13) gives us the sample size of 800. DCM uses this number as a lower bound of the acceptable sample size a domain-concept should have. This means that a domain concept should contain [800, ∞) transactions. If a domain-concept has the number of

transactions less than suggested, DCM generalizes the domain-concept by aggregating (union) the transactions of two or more (*attribute*: *value*) pairs of the same attribute to form a new single domain-concept. For example, if a dc partition with (*age group*: *18-25*) is too small, DCM iteratively aggregates the transactions of (age group: 18-25) with one other age group until the sample size reaches the suggested value. In addition, aggregating dc's of the same attribute still preserves the organization of the data.

In practical settings, DCM uses 800 transactions as a lower bound value for all dc partitions of public health data sets, including the *Centers for Disease Control and Prevention* (CDC)'s *Behavioral Risk Factor Surveillance System* (BRFSS) [36] and the *Agency for Healthcare Research and Quality* (AHRQ)'s *Nationwide Inpatient Sample* (NIS) [35]. The sizes of these data range from 184,450 (BRFSS 2000) to over 8 millions (NIS 2005) transactions; hence, it is relatively seldom that a domain-concept partition with the most granulized criterion of a (*attribute*: *value*) pair would have less than 800 transactions. For other data sets that are a lot smaller than the prior mentioned sets, such as the breast cancer survivor with lymphedema and the synthetic data for industrial engineering machine-group problem, the sample size estimation is not yet implemented. More DCM's applications and implementation details are discussed in Chapter 6.

CHAPTER 4

ON-DEMAND MINING

As discussed previously, the Domain-Concept Mining (DCM) approach groups transactions into domain-concept (dc) partitions. Each dc partition can be mined independently because a dc partition contains all of its dc-qualified transactions. This independency among dc partitions enables DCM to process them offline in either a batch, parallel, or distributed fashion with no estimation or communication among the partitions needed. Thus, the independent dc partitions give DCM an advantage over other association mining approaches that utilize traditional partitioning or sliding window, and parallel or distributed computing techniques. More importantly, frequent itemsets uncovered from each partition by DCM is thorough because the approach implements brute-force association mining algorithms.

After frequent itemsets from each dc partition have been uncovered and stored in a relational database, the next step is to intelligently aggregate the partitions and their frequent itemsets by utilizing information obtained from the offline data mining processes. The objectives are: 1.) not to perform a mining process again because a process is computationally expensive, 2.) allow human experts to utilize the domain-concept organization to combine, compare, and contrast results by adjusting mixtures of domain-concepts, and 3.) obtain a thorough and true set of results by merging domain-

concepts (especially when merging all values of the same attribute) as if the results had been uncovered from the entire data set.

In order to efficiently achieve the above objectives, a novel DCM Partition Aggregation (DCM-PA) approach is developed. DCM-PA utilizes Bayes Theorem [31] along with the database query optimization technique, called pipelining [34]. On the one hand, Bayes Theorem is utilized to compute an actual proportion of the transactions (or support value) that an itemset has when multiple domain-concepts are aggregated. Furthermore, the theorem allows DCM-PA to re-use the information previously obtained from the DCM's offline association mining to achieve the actual support value. On the other hand, the pipelining technique is implemented to maximize the efficiency of the aggregation processes by supplying necessary inputs to the aggregation operators step by step without materializing intermediate results.

The assumptions and/or conditions needed for DCM-PA are:

1.) All (*attribute*: *value*) pairs, any member of their power set (itemsets), and their numbers of transactions (or support values) are known from the DCM partitioning step and offline association mining. Otherwise, their numbers of transactions can be efficiently retrieved from dc partitions based on a reasoning that a partition is significantly smaller than the size of the entire data set, and
2.) All itemsets can be used to partition the data.

Let A_1, A_2, and B be sets of transactions that share a_1 item(s), a_2 item(s), and b item(s), respectively. A Venn diagram, as shown in Figure 4.1 (a), represents these sets as

three domain-concepts. In this example, the aggregation is between the domain-concepts A_1 with a_1 = (*diabetes*: *yes*) and A_2 with a_2 = (*age group*: *65-74*). Furthermore, B with b = (*healthcare coverage*: *no*) is rather considered as a finding instead of a dc partition without the loss of generalization. This implies that item b can be uncovered from the domain-concepts A_1 and A_2 during the DCM offline association mining.

There are two operations offered by DCM-PA to aggregate dc partitions:

1.) A union operation, denoted by $\vee$.

2.) An intersection operation, denoted by $\wedge$.

The operations are to aggregate "transactions" of multiple dc partitions. It is worth mentioning that the operators $\vee$ are $\wedge$ used as opposed to the operators $\cup$ and $\cap$. This is because, according to traditional association mining approaches, operators $\cup$ and $\cap$ are reserved to be used when union or intersect "(*attribute*: *value*) pairs" (columns or itemsets).

Also as shown in Figure 4.1 (b), the union operation of the dc partitions, A_1 and A_2, is to combine all transactions of these dc partitions. The result of this operation is the computed conditional probability of an itemset, b, written in the form of $\mathrm{P}(B \mid A_1 \vee A_2)$. Similarly, Figure 4.1 (c) shows the intersection operation of the transactions from the same dc partitions. The computed conditional probability of b from this intersection operation is denoted by $\mathrm{P}(B \mid A_1 \wedge A_2)$.

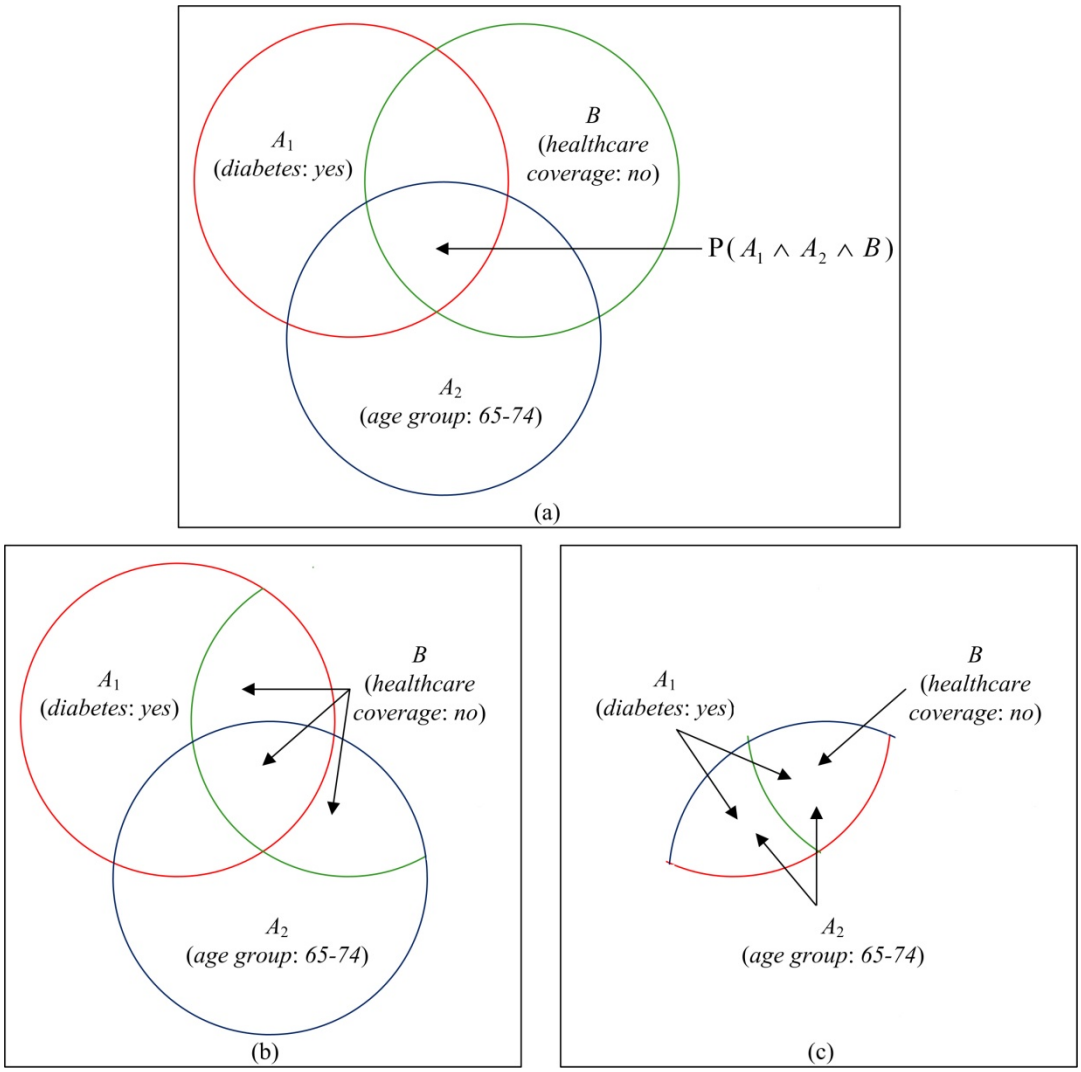


Figure 4.1. (a) A Venn diagram depicts three domain-concept partitions, each of which can be represented by a set of transactions, and their overlapping, (b) a Venn diagram depicts the DCM-PA using the union operation, and (c) a Venn diagram depicts the DCM-PA using the intersection operation.

As already mentioned, an important aspect of the union operation used when aggregates dc partitions is to ultimately achieve the support value of an itemset *b* with respect to all transactions of the original data set. This is to compute the conditional probability:

$$\mathrm{P}\left(B \mid \vee_{i=1}^{n} A_i\right) \tag{4.1}$$

In order to compute the support value of an itemset, regardless of the number of dc partitions aggregated, the set theorem with generalized intersection [31], "Bayes Theorem" [31], basic set properties, and the "Inclusion-Exclusion Principle" [88] through lattices and algebraic structures [89] are utilized.

4.1 Probability and Association Mining

Considering a dc partition A_1, the proportion of the transactions in A_1 with respect to the entire data set is denoted by $\mathrm{P}(A_1)$, and can be calculated by:

$$\mathrm{P}(A_1) = \frac{\text{number of transactions that have } a_1}{\text{total number of transactions in } U} \tag{4.2}$$

, where U represents the whole data set (or the universe of discourse). Please note that the number of transactions that have a_1 is basically the size of the dc partition A_1. The support value of an itemset b in a dc partition A_1 can be written as $\mathrm{P}(B \mid A_1)$, which represents the conditional probability of the itemset b given the dc partition A_1. This value can be calculated by:

$$\mathrm{P}(B \mid A_1) = \frac{\mathrm{P}(A_1 \wedge B)}{\mathrm{P}(A_1)} \tag{4.3}$$

Even though $\mathrm{P}(B \mid A_1)$ is called a support value of b in the dc partition A_1, this same value is also the confidence value of an association rule $a_1 \rightarrow b$. Next is to calculate the proportion of the transactions that have both a_1 and b co-occur at the same time, or $\mathrm{P}(A_1 \wedge B)$, which is the nominator of equation (4.3), by:

$$\mathrm{P}(A_1 \wedge B) = \frac{\text{number of transactions that have both } a_1 \text{ and } b}{\text{total number of transactions in } U} \tag{4.4}$$

However, DCM-PA avoids computing $P(A_1 \wedge B)$ as shown in equation (4.4) because this equation requires a database scan. Instead, DCM-PA is designed to restrict the access to the entire data set especially now that DCM-PA can re-use the information from the offline mining process. Therefore, DCM-PA can obtain $P(A_1 \wedge B)$ more efficiently than equation (4.4) by applying "Multiplication Rule" [31] to equation (4.3):

$$P(A_1 \wedge B) = P(B \mid A_1) \cdot P(A_1) \tag{4.5}$$

Equations (4.2), (4.3), and (4.5) can also be applied to P(B), P($A_1|B$), P(A_2), P(B| A_2), P($A_1|$ A_2), and P($A_2|$ A_1) to obtain $P(A_1 \wedge A_2)$ and $P(A_2 \wedge B)$. Hence, we have all pieces of the needed information to aggregate A_1 and A_2. The details will be discussed in the next section.

4.1.1 Generalized Intersection and Domain-Concept Aggregation

To generalize and simplify the aggregation of multiple dc partitions, we first attempt to find the following conditional probabilities:

$$P(B \mid A_1 \vee A_2) = \frac{P\left(B \wedge (A_1 \vee A_2)\right)}{P(A_1 \vee A_2)} \tag{4.6}$$

and

$$P(B \mid A_1 \wedge A_2) = \frac{P\left(B \wedge (A_1 \wedge A_2)\right)}{P(A_1 \wedge A_2)} \tag{4.7}$$

Let's begin by solving equation (4.6), which is to broaden a partition through a union operation of two dc partitions. From the "General Addition Rule" [31], $P(A_1 \vee A_2)$ can be obtained by:

$$P(A_1 \vee A_2) = P(A_1) + P(A_2) - P(A_1 \wedge A_2) \tag{4.8}$$

Furthermore, the union and intersection operations of sets have a closure property [90]. These set operations also possess the following basic set properties: 1.) associative, 2.) commutative, and 3.) distributive [88]. Therefore, one can apply the distributive property to equation (4.6) to obtain:

$$\begin{aligned} \mathrm{P}(B \mid A_1 \vee A_2) &= \frac{\mathrm{P}\big(B \wedge (A_1 \vee A_2)\big)}{\mathrm{P}(A_1 \vee A_2)} \\ &= \frac{\mathrm{P}\big((B \wedge A_1) \vee (B \wedge A_2)\big)}{\mathrm{P}(A_1 \vee A_2)} \end{aligned} \tag{4.9}$$

Then, equation (4.9) can be expanded using the format of equation (4.8) to obtain:

$$\mathrm{P}(B \mid A_1 \vee A_2) = \frac{\mathrm{P}(B \wedge A_1) + \mathrm{P}(B \wedge A_2) - \mathrm{P}(B \wedge A_1 \wedge A_2)}{\mathrm{P}(A_1) + \mathrm{P}(A_2) - \mathrm{P}(A_1 \wedge A_2)} \tag{4.10}$$

DCM maximizes the aggregation efficiency with the purpose to achieve the solution of $\mathrm{P}(B \mid A_1 \vee A_2)$ without performing a mining process on a new set of transactions, e.g. a set of transactions that have either a_1 or a_2 or both. Therefore, DCM solves equation (4.10) by reusing the following values: 1.) $\mathrm{P}(A_1)$, 2.), $\mathrm{P}(B \wedge A_1)$ 3.), $\mathrm{P}(A_2)$ and 4.) $\mathrm{P}(B \wedge A_2)$. The first two values can be directly obtained from the results of the dc partition A_1, and the latter two can be obtained from the dc partition A_2.

Although, we have not done an offline data mining process on a set of transactions that have both a_1 and a_2, (in other words, $A_1 \wedge A_2$), it is obvious that $\mathrm{P}(A_1 \wedge A_2)$ can be obtained directly from either the dc partition A_1 or A_2. This is simply because $\mathrm{P}(A_1 \wedge A_2)$ can be inferred from the pre-established assumptions (i.e. a_2 is an attribute inside the dc partition A_1, and vice versa). Therefore, the only unknown

component in equation (4.10) is $\mathrm{P}(B \wedge A_1 \wedge A_2)$, which can be obtained by utilizing the following calculations and techniques:

1.) A posteriori probability calculation based on Bayes Theorem. The theorem is detailed step-by-step from equations (4.11) to (4.14) below:

$$\mathrm{P}(B \mid A_1) = \frac{\mathrm{P}(A_1 \wedge B)}{\mathrm{P}(A_1)} \tag{4.11}$$

$$\mathrm{P}(A_1 \wedge B) = \mathrm{P}(A_1) \cdot \mathrm{P}(B \mid A_1) \tag{4.12}$$

$$\mathrm{P}(A_1 \wedge B) = \mathrm{P}(B \wedge A_1) = \mathrm{P}(B) \cdot \mathrm{P}(A_1 \mid B) \tag{4.13}$$

$$\mathrm{P}(B \mid A_1) = \frac{\mathrm{P}(B) \cdot \mathrm{P}(A_1 \mid B)}{\mathrm{P}(A_1)} \tag{4.14}$$

2.) The generalized intersection, which is also detailed step-by-step from equations (4.15) to (4.18), where each step resembles Bayes Theorem above.

$$\mathrm{P}(B \mid A_1 \wedge A_2) = \frac{\mathrm{P}(A_1 \wedge A_2 \wedge B)}{\mathrm{P}(A_1 \wedge A_2)} \tag{4.15}$$

$$\mathrm{P}(A_1 \wedge A_2 \wedge B) = \mathrm{P}(A_1 \wedge A_2) \cdot \mathrm{P}(B \mid A_1 \wedge A_2) \tag{4.16}$$

$$\begin{aligned} \mathrm{P}\big((A_1 \wedge A_2) \wedge B\big) &= \mathrm{P}\big(B \wedge (A_1 \wedge A_2)\big) \\ &= \mathrm{P}(B) \cdot \mathrm{P}\big((A_1 \wedge A_2) \mid B\big) \end{aligned} \tag{4.17}$$

$$\begin{aligned} \mathrm{P}\big(B \mid (A_1 \wedge A_2)\big) &= \frac{\mathrm{P}\big(B \wedge (A_1 \wedge A_2)\big)}{\mathrm{P}(A_1 \wedge A_2)} \\ &= \frac{\mathrm{P}(B) \cdot \mathrm{P}\big((A_1 \wedge A_2) \mid B\big)}{\mathrm{P}(A_1 \wedge A_2)} \end{aligned} \tag{4.18}$$

Therefore, $\mathrm{P}(B \wedge A_1 \wedge A_2)$, which is the only unknown component in equation (4.10), can be obtained by:

$$\mathrm{P}(B \wedge A_1 \wedge A_2) = \mathrm{P}(A_1 \wedge A_2) \cdot \mathrm{P}\left(B \mid (A_1 \wedge A_2)\right) \tag{4.19}$$

because

$$\mathrm{P}\left(B \mid (A_1 \wedge A_2)\right) = \frac{\mathrm{P}(B) \cdot \mathrm{P}\left((A_1 \wedge A_2) \mid B\right)}{\mathrm{P}(A_1 \wedge A_2)} \tag{4.20}$$

Furthermore, all components in equation (4.20) are known from the offline mining processes of the domain-concept A_1, A_2, and B. It is important to note that while we were solving a union operation of two dc partitions, we have solved an intersection operation as shown in equation (4.7) as well. This is because equation (4.20) addresses equation (4.7) straightforwardly. In conclusion, we have successfully aggregated two dc partitions to obtain actual probability values of *b* without performing any further frequent itemset mining process.

The next step is to demonstrate that we can also apply the above solutions when more than two dc partitions are aggregated. The union of all dc partitions, except B, has been established in equation (4.1), which can be further detailed as:

$$\mathrm{P}\left(B \mid \vee_{i=1}^{n} A_i\right) = \frac{\mathrm{P}\left(B \wedge \left(\vee_{i=1}^{n} A_i\right)\right)}{\mathrm{P}\left(\vee_{i=1}^{n} A_i\right)} \tag{4.21}$$

According to the set closure property [90], we can substitute $(\vee_{i=1}^{n} A_i)$ component in (4.21) with a set C to obtain the following.

$$\mathrm{P}\left(B \mid \vee_{i=1}^{n} A_i\right) = \mathrm{P}(B \mid C) = \frac{\mathrm{P}(B \wedge C)}{\mathrm{P}(C)} \tag{4.22}$$

Hence, the problem of the union of all (or multiple) dc partitions can be reduced to the problem of the union of two dc partitions. Similarly, the problem of the intersection of multiple or all dc partitions can also be reduced the same way because of the same reason.

The set properties, which include closure, associative, distributive, and commutative, are important. This is because they allow DCM-PA to strictly utilize only the information that has been previously obtained; hence, DCM-PA can achieve a high efficiency. Moreover, DCM-PA does not store intermediate results because they could be redundant and would result in high computational costs. The intermediate results if they were stored is called "materialized" [34], which would add the costs of storage, writing, and reading them back for use. On the contrary, DCM-PA can be efficient by calculating needed probabilities through Bayes Theorem, which is a "pipeline" [34] of processes. By using Bayes Theorem, DCM-PA accumulatively infers or propagates the calculations until it finally achieves the final solution. Hence, the term "on-demand" (or equivalently called "on-the-fly" in [34]) is used. DCM pipeline processes can be illustrated in Figure 4.2. More details will also be discussed in subsequent sections.

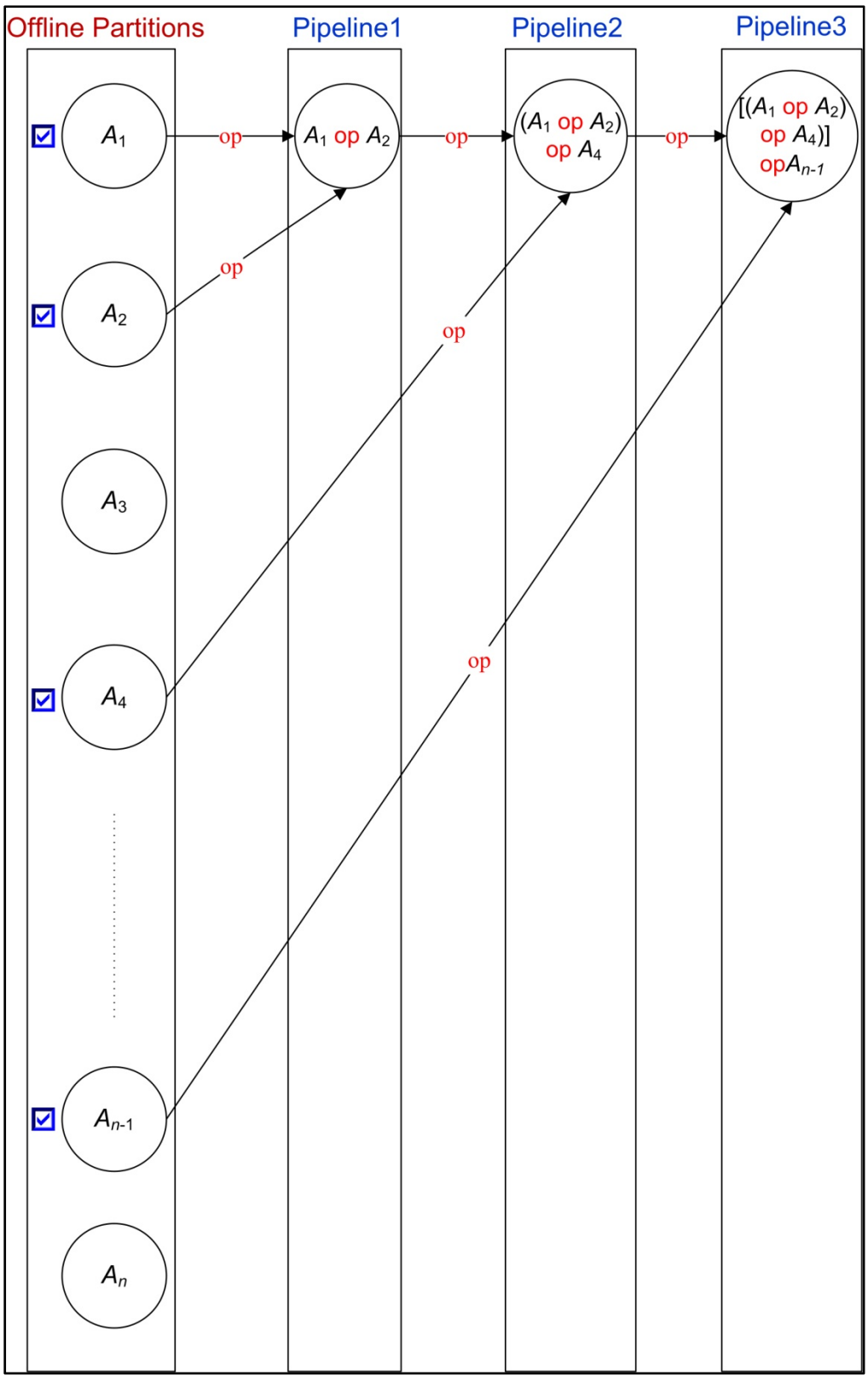


Figure 4.2. On-demand DCM-PA and its pipeline processes through Bayes Theorem, where "op" is either union ($\vee$) or intersection ($\wedge$) operation.

4.1.2 An Alternative Solution by Using Generalized Union

The union operation on a two-set problem and its probability calculation has been previously detailed in equation (4.10). To further clarify equation (4.21), which contains a generalized set union (a series of the union operations of more than two sets), one can formulate a union operation among three sets [91] as:

$$\begin{aligned}
\mathrm{P}(A_1 \vee A_2 \vee A_3) &= \mathrm{P}\left(A_1 \vee (A_2 \vee A_3)\right) \\
&= \mathrm{P}(A_1) + \mathrm{P}(A_2 \vee A_3) - \mathrm{P}\left[A_1 \wedge (A_2 \vee A_3)\right] \\
&= \mathrm{P}(A_1) + \left[\mathrm{P}(A_2) + \mathrm{P}(A_3) - \mathrm{P}(A_2 \wedge A_3)\right] - \mathrm{P}\left[(A_1 \wedge A_2) \vee (A_1 \wedge A_3)\right] \\
&= \mathrm{P}(A_1) + \mathrm{P}(A_2) + \mathrm{P}(A_3) - \mathrm{P}(A_2 \wedge A_3) - \\
&\quad \left\{\mathrm{P}(A_1 \wedge A_2) + \mathrm{P}(A_1 \wedge A_3) - \mathrm{P}\left[(A_1 \wedge A_2) \wedge (A_1 \wedge A_3)\right]\right\} \\
&= \mathrm{P}(A_1) + \mathrm{P}(A_2) + \mathrm{P}(A_3) - \mathrm{P}(A_1 \wedge A_2) - \mathrm{P}(A_1 \wedge A_3) - \mathrm{P}(A_2 \wedge A_3) + \\
&\quad \mathrm{P}(A_1 \wedge A_2 \wedge A_3)
\end{aligned} \tag{4.23}$$

The property of equation (4.23) for n sets is also known as the "Inclusion-Exclusion Principle" [88], which details the following:

$$\begin{aligned}
\mathrm{P}\left(\vee_{i=1}^{n} A_i\right) &= \sum\nolimits_{i=1}^{n} \mathrm{P}\left(A_i\right) \\
&\quad - \sum_{1 \le i < j \le n} \mathrm{P}\left(A_i \wedge A_j\right) \\
&\quad + \sum_{1 \le i < j < k \le n} \mathrm{P}\left(A_i \wedge A_j \wedge A_k\right) \\
&\quad - \ldots + (-1)^{n-1} \mathrm{P}\left(\wedge_{i=1}^{n} A_i\right)
\end{aligned} \tag{4.24}$$

The Inclusion-Exclusion Principle can be computed efficiently, as can equation (4.23). It is worth mentioning that this principle resembles a well-known statistical trial, called Bernoulli that forms the Bernoulli distribution [31], which is a foundation of the Binomial and Normal distributions.

However, a very important discussion regarding the use of this principle directly to obtain either $\mathrm{P}\left(B \mid \vee_{i=1}^{n} A_i\right)$ or $\mathrm{P}\left(B \mid (\vee_{i=1}^{n} A_i) \vee A_{n+1}\right)$ may be plausible only under the following assumption:

All members and their numbers of transactions of the following sets are known:

$$\{(B \wedge A_1), (B \wedge A_2), ..., (B \wedge A_{n+1}),$$
$$(B \wedge A_1 \wedge A_2), ..., (B \wedge A_n \wedge A_{n+1}),$$
$$...,$$
$$(B \wedge A_1 \wedge A_2, ..., A_n \wedge A_{n+1})\}$$

Practically, however, computational costs of maintaining each member's number of transactions to fulfill equation (4.24) could be expensive because all set intersections and their redundancies must be stored (or materialized) as intermediate results. In addition, equation (4.24) requires multiple database scans (or queries), each of which is to retrieve the probability value of an intersection. Moreover, this equation does not strictly pursue the purpose of DCM-PA, which is to intelligently reuse the associations discovered from the DCM offline mining processes. This is because DCM-PA's foundation is its efficiency that could be obtained by inferring or computing a needed conditional probability value. Therefore, an intersection operation, such as $(B \wedge A_1 \wedge A_2, ..., A_n \wedge A_{n+1})$ to obtain $\mathrm{P}\left(B \mid (\wedge_{i=1}^{n} A_i) \wedge A_{n+1}\right)$ is also not preferred; although, it is possible through a database query. For readers who are interested in more investigations of the Inclusion-Exclusion Principle when applied to calculate $\mathrm{P}\left(B \mid (\vee_{i=1}^{n} A_i) \vee A_{n+1}\right)$ value, the proof can be found in Appendix C.

4.1.3 Domain-Concept Partition Aggregation (DCM-PA) On-Demand

This section is to illustrate the DCM-PA approach step-by-step strictly based on the set theorem and Bayes Theorem [31]. Before we detail the DCM-PA approach further, the followings are the basis of the discussion:

1.) All domain-concepts (A_i)have been mined offline separately,
2.) The results of the offline mining are stored in a database, and they are organized according to their domain-concepts.
3.) Users then select sets of interesting domain-concepts they are interested in seeing the aggregated findings,
4.) *b* is an itemset that represents a finding, which is expected to be discovered from this on-demand mining approach. *b*'s set of transactions is represented by *B*.

4.1.3.1 Intersect Multiple Domain-Concept Partitions

DCM-PA intersection ability is offered to the users so that they can narrow the partitioning criteria while viewing the DCM associations. For example, an association reported to the users from the domain-concept (*diabetes*: *yes*) is (*BMI*: *overweight*). Suppose the users would like to investigate further if this association is still true with a group of people who not only have been told they have diabetes, but also: 1.) do not have a healthcare coverage, or the domain-concept (*healthcare coverage*: *no*), and 2.) those who reported they would also say that their general health is poor, or the domain-concept (*general health*: *poor*). The users can select these other two domain-concepts to aggregate with the original domain-concept (*diabetes*: *yes*).

As discussed briefly in Section 4.1.1 regarding the DCM-PA's pipeline processes (see also Figure 4.2), in this section, the discussion will lay out the processes step-by-step. The objective of DCM-PA is to calculate the following to achieve the aggregation result.

$$\mathrm{P}\left(B \mid \left(\wedge_{i=1}^{n} A_i\right)\right) = \frac{\mathrm{P}\left(B \wedge \left(\wedge_{i=1}^{n} A_i\right)\right)}{P\left(\wedge_{i=1}^{n} A_i\right)}, \forall B, A_i \in U, n \geq 1 \tag{4.25}$$

, where U is a set of all transactions in a data set D. Let $\mathrm{P}(B)$ and $\mathrm{P}(B \wedge A_i)$ be the probability values of an itemset b and an itemset (b, a_i), respectively. The most basic case of equation (4.25) is the probability value of an itemset b from a domain-concept A_1, as shown in the following equation:

$$\mathrm{P}(B \mid A_1) = \frac{\mathrm{P}(B \wedge A_1)}{\mathrm{P}(A_1)} \tag{4.26}$$

To aggregate A_1 and A_2 by using the intersection operation is to calculate the conditional probability of:

$$\mathrm{P}(B \mid A_1 \wedge A_2) = \frac{\mathrm{P}\left(B \wedge (A_1 \wedge A_2)\right)}{\mathrm{P}(A_1 \wedge A_2)} \tag{4.27}$$

, where the distributive property [88] can be applied to the $\wedge$ operation. Therefore, equation (4.27) can be re-written as:

$$\mathrm{P}(B \mid A_1 \wedge A_2) = \frac{\mathrm{P}\left((B \wedge A_1) \wedge (B \wedge A_2)\right)}{\mathrm{P}(A_1 \wedge A_2)} \tag{4.28}$$

, where

$$\left((B \wedge A_1) \wedge (B \wedge A_2)\right) = (B \wedge A_1 \wedge A_2) \tag{4.29}$$

Then, Bayes Theorem is applied to equation (4.29) to infer the value of $\mathrm{P}(B \wedge A_1 \wedge A_2)$ from their a priori probabilities (as shown in equation (4.19) in Section 4.1.1).

Next is to expand the pattern in equation (4.27) to an aggregation of $A_1 \wedge A_2 \wedge \ldots \wedge A_n$. Therefore, equation (4.27) can be re-written as:

$$\begin{aligned} \mathrm{P}(B \mid A_1 \wedge A_2 \wedge \ldots \wedge A_n) &= \frac{\mathrm{P}\left(B \wedge (A_1 \wedge A_2 \wedge \ldots \wedge A_n)\right)}{\mathrm{P}\left(A_1 \wedge A_2 \wedge \ldots \wedge A_n\right)} \\ &= \frac{\mathrm{P}\left(B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_n\right)}{\mathrm{P}\left(A_1 \wedge A_2 \wedge \ldots \wedge A_n\right)} \end{aligned} \tag{4.30}$$

, where the denominator is a sub-problem of the nominator. Therefore, it is reasonable to simplify the problem to solve the value of the nominator. Then, one can applied the "Multiplication Rule" [31] to the nominator of equation (4.30) as shown below.

$$\begin{aligned} \mathrm{P}\left(B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_n\right) = \mathrm{P}(B) &\cdot \mathrm{P}(A_1 \mid B) \cdot \mathrm{P}(A_2 \mid B \wedge A_1) \\ &\cdot \mathrm{P}(A_3 \mid B \wedge A_1 \wedge A_2) \cdot \ldots \\ &\cdot \mathrm{P}(A_n \mid B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_{n-1}) \end{aligned} \tag{4.31}$$

,where

$$\mathrm{P}(B \wedge A_1) \;=\; \mathrm{P}(B) \cdot \mathrm{P}(A_1 \mid B) \tag{4.32}$$

$$\mathrm{P}(B \wedge A_1 \wedge A_2) \;=\; \mathrm{P}(B) \cdot \mathrm{P}(A_1 \mid B) \cdot \mathrm{P}(A_2 \mid B \wedge A_1) \tag{4.33}$$

, and so on. Hence, the problem can be pipelined as follow:

$$\mathrm{P}\left(B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_n\right) = \mathrm{P}\left((((B \wedge A_1) \wedge A_2) \wedge \ldots) \wedge A_n\right) \tag{4.34}$$

Specifically, Bayes Theorem is applied to each of the $\mathrm{P}(x \mid \tau) \mid \tau = (p \wedge q \wedge \ldots \wedge r)$ in equation (3.1) when τ is composed of at least two sets intersection. This is to infer the

posteriori probability, e.g. $\mathrm{P}(A_n \mid B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_{n-1})$, from the known (or a priori) probabilities. This is because there has not been an aggregation or an intermediate result of $(B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_{n-1})$ materialized previously. Therefore, Bayes Theorem is used as follow:

$$\mathrm{P}(A_n \mid B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_{n-1}) = \frac{\mathrm{P}(A_n \wedge B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_{n-1})}{\mathrm{P}(B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_{n-1})} \tag{4.35}$$

$$\mathrm{P}(B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_{n-1} \mid A_n) = \frac{\mathrm{P}(B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_n)}{\mathrm{P}(A_n)} \tag{4.36}$$

$$\mathrm{P}(B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_n) = \mathrm{P}(B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_{n-1} \mid A_n) \cdot \mathrm{P}(A_n) \tag{4.37}$$

$$\mathrm{P}(A_n \mid B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_{n-1}) = \frac{\mathrm{P}(B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_{n-1} \mid A_n) \cdot \mathrm{P}(A_n)}{\mathrm{P}(B \wedge A_1 \wedge A_2 \wedge \ldots \wedge A_{n-1})} \tag{4.38}$$

The denominator value of the above equation has been calculated prior this equation because of pipelining. In conclusion, the calculation of $\mathrm{P}\left(B \mid \left(\wedge_{i=1}^{n} A_i\right)\right), n \geq 1$ can be done accumulatively, two sets at a time. And, the probability values needed for each step are known or can be inferred by using Bayes Theorem.

4.1.3.2 Union Multiple Domain-Concept Partitions

A union of multiple domain-concepts is important because it can be utilized for cases include: 1.) incremental data mining, 2.) compare and contrast the effect of numbers of transactions. For the first case, dc partitions can be a partition of current data, and another of new data. After DCM has mined both partitions for associations, DCM-PA performs a union operation to combine the transactions of both partitions together. Hence, the current associations are updated by the new findings.

An example of the second case is a set of findings that could be verified from trivial knowledge or beliefs. For example, one may believe that there is an association between a particular disease and overweight or obesity. Suppose a finding uncovered from the domain-concept (*diabetes*: *yes*) shows that (*BMI*: *overweight*) has a high support value, which could partially confirm the prior knowledge of the experts. However, by combining domain-concepts (*diabetes*: *yes*) and (*diabetes*: *no*) together through the DCM-PA union operation shows that (*BMI*: *overweight*) may have a lower support value, but the value is still higher than the threshold. This piece of information could be useful for the experts' further validations of their beliefs to find out whether overweight is in fact an epidemic problem of a general population or a specific problem associated with people with diabetes.

The DCM pipelining process for the union operation starts from the offline associations that are organized based on their domain-concepts are stored in a database as shown in Figure 4.2. A Web interface, called DCMiner (which will also be explained further in Chapter 6), allows the users to select domain-concepts to perform a union operation. In this case, the selected domain-concepts are A_1, A_2, A_4, and $A_{n\text{-}1}$. Without the loss of generalization, let b be a set of items that would be the end results of the aggregation operation, B be a set of transactions that have b, A_i re-index the selected domain-concept to be i = 1 to 4, and U be a set of all transactions in a data set D. Therefore, DCM-PA performs the following calculation:

$$\mathrm{P}\left(B \mid \left(\vee_{i=1}^{n} A_i\right)\right) = \frac{\mathrm{P}\left(B \wedge \left(\vee_{i=1}^{n} A_i\right)\right)}{P\left(\vee_{i=1}^{n} A_i\right)} \Bigg| \forall B, A_i \in U, n = 4 \tag{4.39}$$

The goal is to show that the above equation will result in the correct probability value of an item *b* after DCM-PA aggregates these multiple A_i partitions through pipelining.

Let P(B) and $\mathrm{P}(B \wedge A_i)$ be the probability values of an itemset *b* and an itemset (*b*, a_i), respectively. DCM-PA starts with the probability of the finding in the first domain-concept, or $\mathrm{P}(B \mid A_1)$ as shown below.

$$\mathrm{P}(B \mid A_1) = \frac{\mathrm{P}(B \wedge A_1)}{\mathrm{P}(A_1)} \tag{4.40}$$

Then, DCM-PA proceeds to its first aggregation between A_1 and A_2, which is to calculate the conditional probability of:

$$\mathrm{P}(B \mid A_1 \vee A_2) = \frac{\mathrm{P}\big(B \wedge (A_1 \vee A_2)\big)}{\mathrm{P}(A_1 \vee A_2)} \tag{4.41}$$

The distributive property [88] can be applied to both of the $\vee$ and $\wedge$ operations. Hence, we obtain:

$$\mathrm{P}(B \mid A_1 \vee A_2) = \frac{\mathrm{P}\big((B \wedge A_1) \vee (B \wedge A_2)\big)}{\mathrm{P}(A_1 \vee A_2)} \tag{4.42}$$

, which is re-arranged to be a problem of sets union, which is defined in the "General Addition Rule" [31] as follow:

$$\mathrm{P}(A_1 \vee A_2) = \mathrm{P}(A_1) + \mathrm{P}(A_2) - \mathrm{P}(A_1 \wedge A_2) \tag{4.43}$$

Hence, one can apply equation (4.43) to equation (4.42) to get:

$$\mathrm{P}(B \mid A_1 \vee A_2) = \frac{\mathrm{P}(B \wedge A_1) + \mathrm{P}(B \wedge A_2) - \mathrm{P}(B \wedge A_1 \wedge A_2)}{\mathrm{P}(A_1) + \mathrm{P}(A_2) - \mathrm{P}(A_1 \wedge A_2)} \tag{4.44}$$

Furthermore, all components on the right hand side of the above are known, except $\mathrm{P}(B \wedge A_1 \wedge A_2)$, which can instead be inferred using Bayes Theorem as discussed previously in Section 4.1.1. Also, without the loss of generalization, both of the nominator and denominator of equation (4.44) are calculated the same way, and B is implied to be the expected finding from the selected domain-concepts. Therefore, the problem can be reduced to $\mathrm{P}(A_1 \vee A_2)$.

The next DCM-PA step is to pipeline the rest of the selected domain-concept into the calculation as follow:

$$\mathrm{P}\left(\vee_{i=1}^{n} A_i\right) = \mathrm{P}(((A_1 \vee A_2) \vee A_3) \vee A_4) \tag{4.45}$$

This calculation is possible because sets union and intersection possess the closure, associative, commutative, and distributive properties [90].

In conclusion, the problem of multiple domain-concepts union is reduced to a union of two sets at a time. It is also worth mentioning that, based on the DCM-PA pipeline processes, the aggregation through a series of union operations can be applied to any number of selected domain-concept partitions (i.e. $n \geq 1$).

4.1.3.3 Union Multiple Domain-Concept Partitions for Complete Results

As discussed in the previous sections regarding the calculations of the DCM-PA union operation, this section details how DCM-PA obtains a set of complete results. The step-by-step DCM-PA discussed recently shows that DCM-PA has the ability to aggregate dc partitions that may have different sizes and overlap each other (see also

Figure 4.1). Therefore, an aggregation of all dc partitions that share the *same attribute* will lead to a merging of all transactions of the data set together; and thus, DCM-PA can obtain a set of complete results with respect to the attribute. Further, a set of all dc partitions that share the same attribute has no overlapping transactions among the dc partitions. In other words, these partitions are mutually exclusive. According to the Set Addition Principle, an aggregation of these dc partitions implies the following equation:

$$\mathrm{P}(\vee_{i=1}^{n} A_i) = \sum_{i=1}^{n} \mathrm{P}(A_i) \tag{4.46}$$

For example, as shown in Figure 4.3, an attribute "age group" has 6 different values. The (*attribute*: *value*) pairs for age group are (*age group*: *18-24*), (*age group*: *25-34*), (*age group*: *35-44*), (*age group*: *45-54*), (*age group*: *55-64*), and (*age group*: *65 or older*). In addition, a transaction that qualifies for one dc partition, such as (*age group*: *18-24*), will not qualify for the other age group partitions. Consequently, DCM-PA can utilize the union operation to aggregate all age groups' partitions in order to: 1) merge all transactions in the data set, and 2.) obtain the complete set of results that are associated to the overall concept of age group *without* processing an offline data mining process for the entire data set.

This aggregation of a set of domain-concepts with the same attribute can be compared directly to traditional association mining approaches, which mine the entire set of transactions without partitions. A main disadvantage that the traditional approaches have is that it may take a considerable amount of time to process, and it may not be able to achieve any result at all if 1.) the support value is set high (so that the process can be completed or completed quicker), or 2.) the process has used up all of the computational

resources available. Further, the results from the traditional approach not organized; hence it is not easy to browse or make comparisons. Further details, experimentations, and comparisons between DCM with DCM-PA and the traditional approach can be found in Chapter 5.

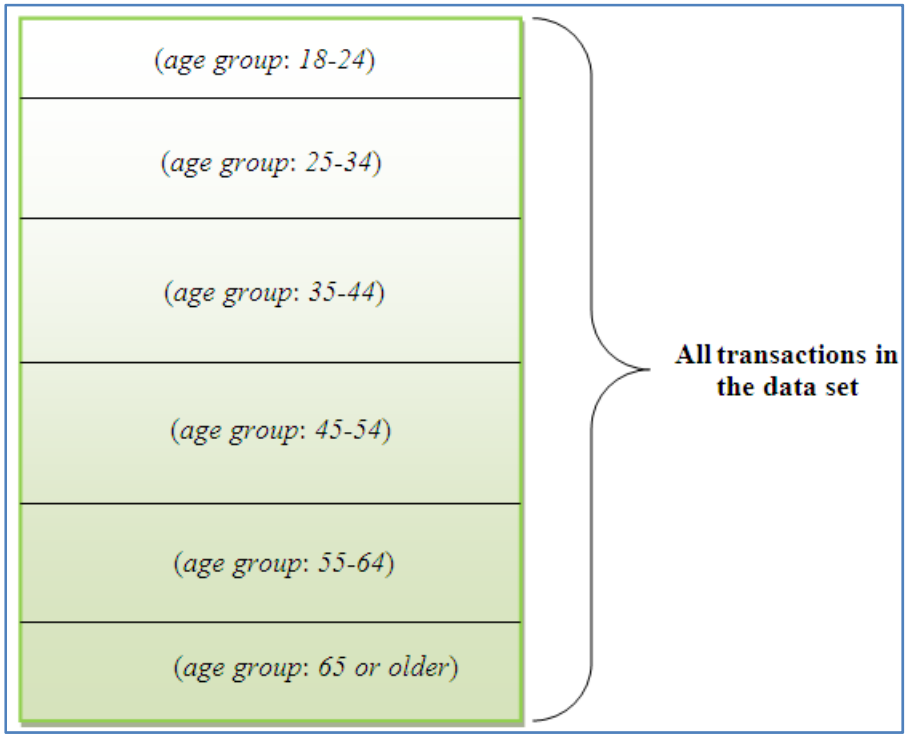


Figure 4.3. DCM-PA union operations for all age group partitions.

4.1.4 Domain-Concept Partition Aggregation with Negations and Mixtures of Union and Intersection Operations

This section details the other aggregation situations that the DCM-PA approach can be applied. The aggregation situations include: 1.) a domain-concept and its negation, and 2.) a mixture of the union and intersection operations.

4.1.4.1 Aggregation between Domain-Concept and Its Negation

The problem of aggregating a domain-concept and its negation (denoted by $\neg$) can be illustrated in Figure 4.4.

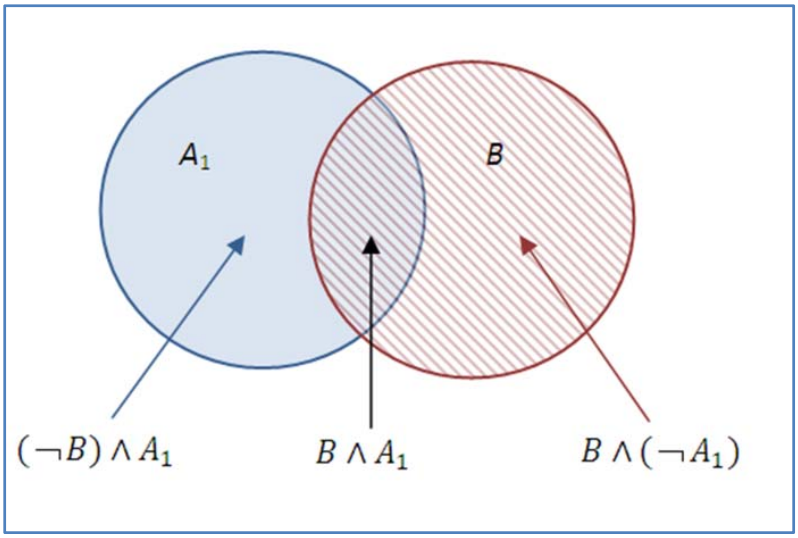


Figure 4.4. A Venn diagram represents various set intersection situations.

The first problem is to calculate the value of the following.

$$\mathrm{P}(B \mid A_1 \vee (\neg A_1)) = \frac{\mathrm{P}\left((B \wedge A_1) \vee \left(B \wedge (\neg A_1)\right)\right)}{\mathrm{P}\left(A_1 \vee (\neg A_1)\right)} \tag{4.47}$$

, which is equivalent to:

$$\mathrm{P}(B \mid A_1 \vee (\neg A_1)) = \frac{\mathrm{P}(B \wedge A_1) + \mathrm{P}(B \wedge (\neg A_1))}{\mathrm{P}(U)} = \frac{\mathrm{P}(B)}{\mathrm{P}(U)} = \mathrm{P}(B) \tag{4.48}$$

This is because the nominator of equation (4.47) is disjoint, and P(U) is 1, where U is the entire data set. In general, $\mathrm{P}(Y \mid X \vee (\neg X)) = \mathrm{P}(Y)$, where X and Y can be multiple sets.

The application of this kind of aggregation is the same as a union of multiple domain-concepts discussed in Sections 4.1.3.2 and 4.1.3.3. In other words, a union of a

domain-concept and its negation is the same as a union of the set of domain-concepts that have the same attribute. For a general case of negation that involves various attributes, e.g. a (broader) domain-concept that is form by an aggregation between (*age group*: *18-24*) and (*healthcare coverage*: *no*), its negation can also be calculated the same way. That is to merge all domain-concepts that their attributes are "age group" or "healthcare coverage".

For a purpose of explorations, the second problem, which is rather unusual, is to calculate the value of the following.

$$\begin{aligned} \mathrm{P}\left(B \mid A_1 \wedge (\neg A_1)\right) &= \frac{\mathrm{P}\left((B \wedge A_1) \wedge \left(B \wedge (\neg A_1)\right)\right)}{\mathrm{P}\left(A_1 \wedge (\neg A_1)\right)} \\ &= \frac{\mathrm{P}\left(B \wedge A_1 \wedge (\neg A_1)\right)}{\mathrm{P}\left(A_1 \wedge (\neg A_1)\right)} \end{aligned} \tag{4.49}$$

, which is not computable due to the values of $\mathrm{P}\left(B \wedge A_1 \wedge (\neg A_1)\right)$ and $\mathrm{P}\left(A_1 \wedge (\neg A_1)\right)$ are both zero. Therefore, any $\mathrm{P}(Y \mid X \wedge (\neg X))$ is not applicable.

4.1.4.2 Aggregation Using Mixtures of Union and Intersection Operations

An example of the problem of a mixture of the union and intersection operations can be detailed as follows.

$$\mathrm{P}\left(B \mid \left((A_1 \vee A_2) \wedge A_3\right)\right) = \frac{\mathrm{P}\left(\left(B \wedge (A_1 \wedge A_3)\right) \vee (B \wedge A_2 \wedge A_3)\right)}{\mathrm{P}\left(\left(A_1 \wedge A_3\right) \vee \left(A_2 \wedge A_3\right)\right)} \tag{4.50}$$

Both components of the nominator can be calculated by using Bayes Theorem as previously explained in Section 4.1.3.1. Then, the Set Addition Principle is utilized to

achieve the value of the nominator. For the denominator, both components can be obtained directly from offline mining results, and then the Set Addition Principle is applied. In general, the problem can be re-arranged to a disjunctive normal form (DNF) [34], which is a union of multiple domain-concepts as follows.

$$\begin{aligned} & \mathrm{P}\left(B \mid \left((\vee_{i=1}^{n} A_i) \wedge A_j\right)\right) \\ & = \frac{\mathrm{P}\left(\left(B \wedge (A_1 \wedge A_j)\right) \vee (B \wedge (A_2 \wedge A_j)) \vee \ldots \vee (B \wedge (A_n \wedge A_j))\right)}{\mathrm{P}\left(\left(A_1 \wedge A_j\right) \vee \left(A_2 \wedge A_j\right) \vee \ldots \vee \left(A_n \wedge A_j\right)\right)} \\ & = \frac{\mathrm{P}\left(\vee_{i=1}^{n} \left(B \wedge A_i \wedge A_j\right)\right)}{\mathrm{P}\left(\vee_{i=1}^{n} \left(A_i \wedge A_j\right)\right)} \end{aligned} \tag{4.51}$$

Without the loss of generalization, B can be omitted for a purpose of a simpler illustration. Therefore, an on-demand aggregation problem can be pipelined as illustrated in Figure 4.5. Please note that each of the intersection processes of the offline partitions may be done when it is needed to be used in the pipeline, e.g. all of the intersections do not need to be computed at the same time. This is to maximize the efficiency by minimizing the use of the main memory.

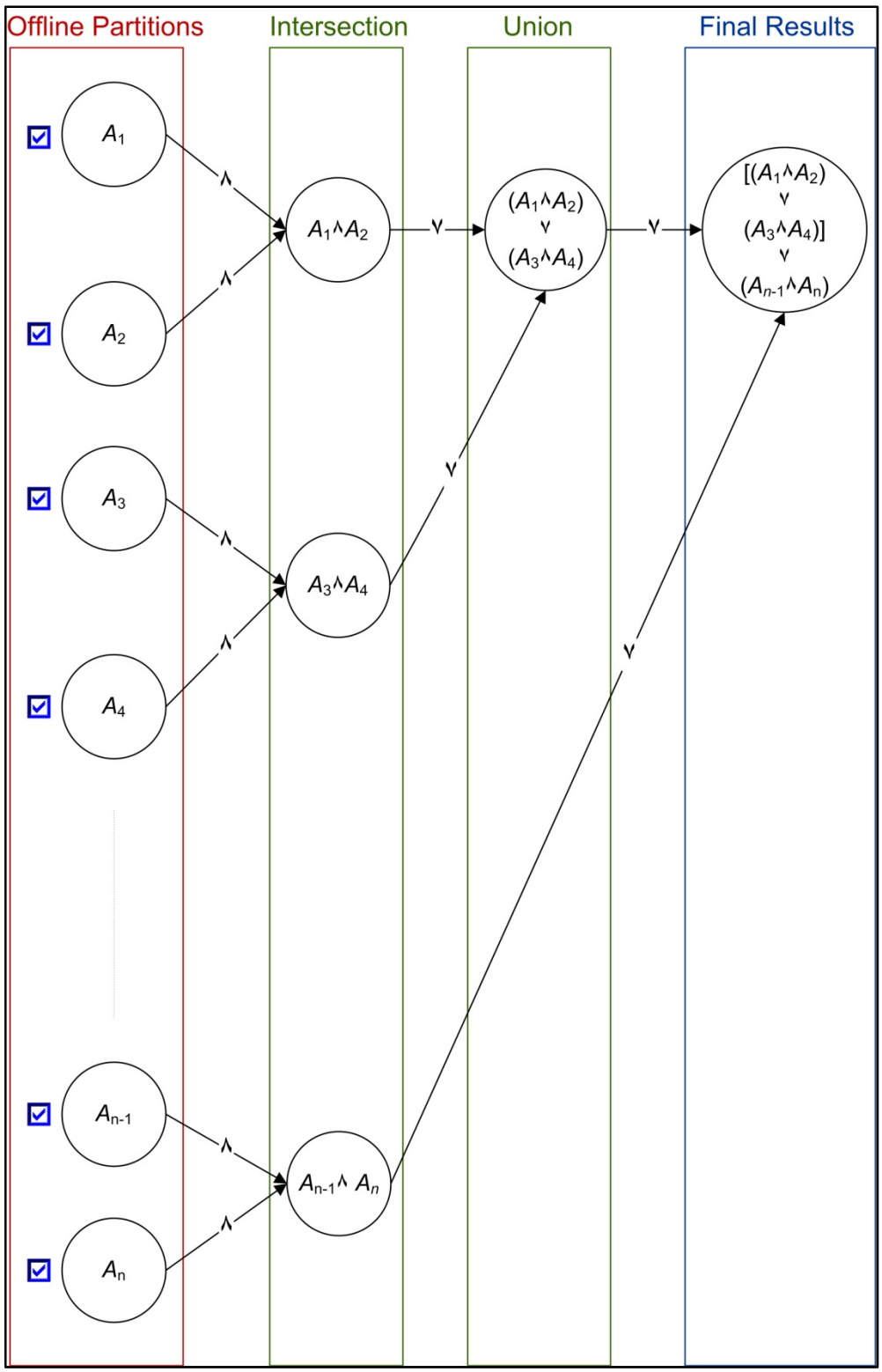


Figure 4.5. Pipeline processes of a mixture between intersection and union operations in a disjunctive normal form.

The next problem is the intersection and then the union operations of multiple domain-concepts. It can be formulated as:

$$\begin{aligned} \mathrm{P}\left(B \mid \left((A_1 \wedge A_2) \vee A_3\right)\right) &= \frac{\mathrm{P}\left(B \wedge \left((A_1 \wedge A_2) \vee A_3\right)\right)}{\mathrm{P}\left(\left(A_1 \vee A_3\right) \wedge \left(A_2 \vee A_3\right)\right)} \\ &= \frac{\mathrm{P}\left(\left(B \wedge (A_1 \vee A_3)\right) \wedge \left(B \wedge (A_2 \vee A_3)\right)\right)}{\mathrm{P}\left(\left(A_1 \vee A_3\right) \wedge \left(A_2 \vee A_3\right)\right)} \\ &= \frac{\mathrm{P}\left(\left((B \wedge A_1) \vee (A_1 \wedge A_3)\right) \wedge \left((B \wedge A_2) \vee (B \wedge A_3)\right)\right)}{\mathrm{P}\left(\left(A_1 \vee A_3\right) \wedge \left(A_2 \vee A_3\right)\right)} \end{aligned} \tag{4.52}$$

In general, the problem can be re-arranged to a conjunctive normal form (CNF) [34] as follows.

$$\begin{aligned} &\mathrm{P}\left(B \mid \left((\wedge_{i=1}^{n} A_i) \vee A_j\right)\right) \\ &= \frac{\mathrm{P}\left(\left(B \wedge (A_1 \vee A_j)\right) \wedge (B \wedge (A_2 \vee A_j)) \wedge \ldots \wedge (B \wedge (A_n \vee A_j))\right)}{\mathrm{P}\left(\left(A_1 \vee A_j\right) \wedge \left(A_2 \vee A_j\right) \wedge \ldots \wedge \left(A_n \vee A_j\right)\right)} \\ &= \frac{\mathrm{P}\left(B \wedge \left(\wedge_{i=1}^{n} (A_i \vee A_j)\right)\right)}{\mathrm{P}\left(\wedge_{i=1}^{n} \left(A_i \vee A_j\right)\right)} \end{aligned} \tag{4.53}$$

, which can be solved by pipelining as well. Without the loss of generalization, B is omitted. Therefore, an on-demand aggregation process is illustrated in Figure 4.6.

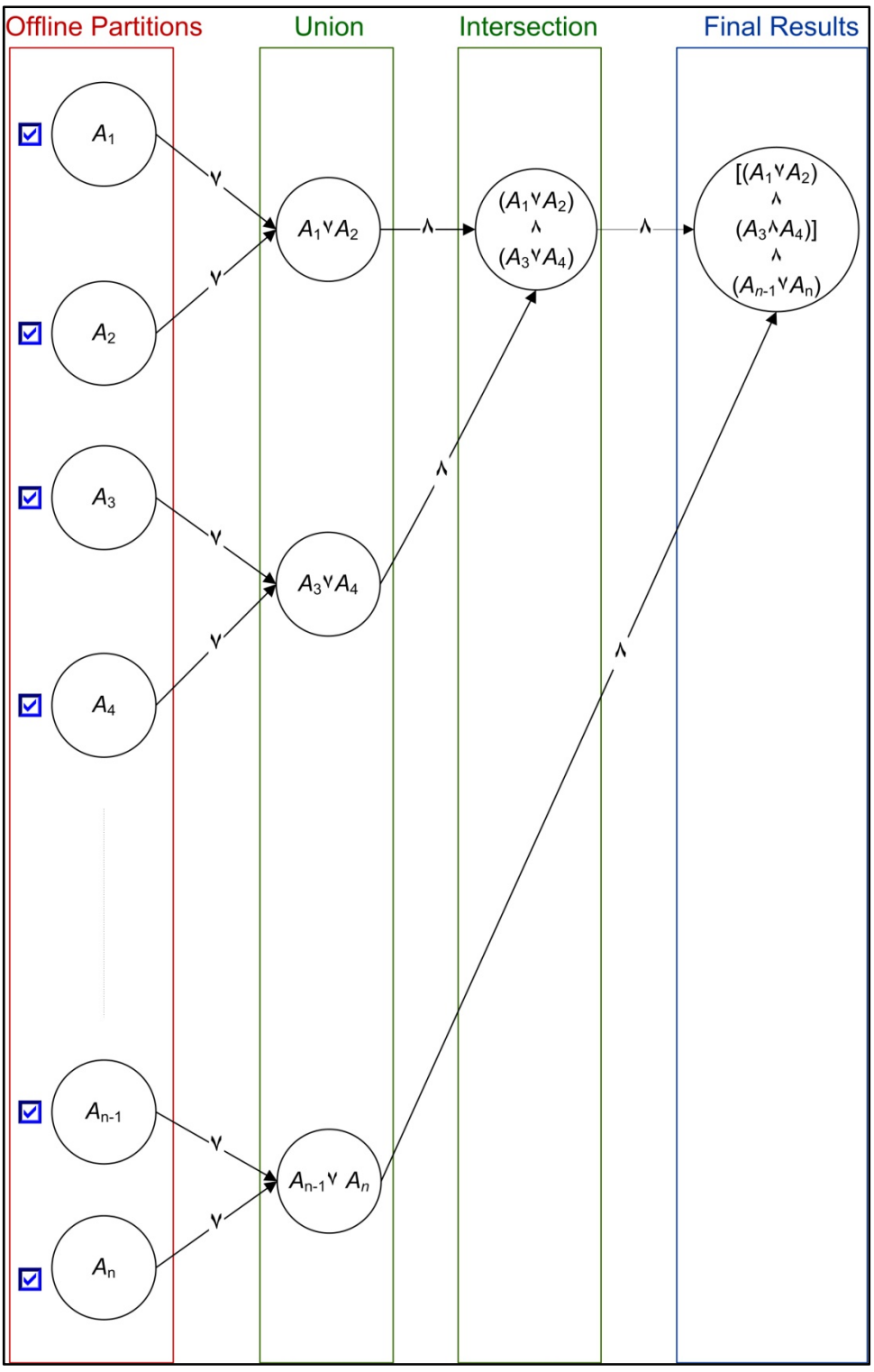


Figure 4.6. Pipeline processes of a mixture between intersection and union operations in a conjunctive normal form.

CHAPTER 5

EXPERIMENTAL RESULTS AND EVALUATION

The data set used in the evaluation of the DCM and DCM-PA approaches is from the "Behavioral Risk Factor Surveillance System" (BRFSS) 2006 [33] survey data. BRFSS 2006 contains 355,710 transactions with 302 variables that are collected from a stratified random sample of adults (age 18 years or older), with a maximum of one adult surveyed per household. The samples are drawn throughout 50 states, the District of Columbia, Puerto Rico, Guam, and the Virgin Islands. The BRFSS employed a telephone survey method to collect the health related data, which included data on behavioral risk factors and health practices that are related to chronic diseases, injuries, and some infectious diseases. The following is sample of survey questions (from 302 originally collected questions and variables) and corresponding answers:

1. "Have you ever been told by a doctor that you have diabetes?" (If "Yes" and respondent is female, as "Was this only when you were pregnant?". If respondent answered "pre-diabetes or borderline diabetes", response code 4 was used.)
   1.1. Value: 1. Value Label: Yes.
   1.2. Value: 2. Value Label: Yes, but female told only during pregnancy.
   1.3. Value: 3. Value Label: No.

1.4. Value: 4. Value Label: No, pre-diabetes or borderline diabetes.

1.5. Value: 7. Value Label: Don't know/Not sure.

1.6. Value: 9. Value Label: Refused.

1.7. Value: BLANK. Value Label: Not asked or Missing.

2. "Has a doctor, nurse, or other health professional ever told you that you had a heart attack, also called a myocardial infarction?"

2.1. Value: 1. Value Label: Yes.

2.2. Value: 2. Value Label: No.

2.3. Value: 7. Value Label: Don't know/Not sure.

2.4. Value: 9. Value Label: Refuse.

Derived domain-concepts from the above survey questions and their answers:

1. (*DIABETE2*: 1), which corresponds to question 1 with choice 1.1.
2. (*DIABETE2*: 3), which corresponds to question 1 with choice 1.3.
3. (*CVDINFR*: 1), which corresponds to question 2 with choice 2.1.
4. (*CVDINFR*: 2), which corresponds to question 2 with choice 2.2.

As detailed in Table Appendix A.1, there are 84 domain-concepts selected from 302 variables of the BRFSS 2006 data set for testing purposes. Unselected BRFSS variables were classified as identifiers, non-core questions that were not asked consistently every year, geographic variables included area codes, stratum codes used for statistical purposes, calculated variables, and the values "Don't know/Not Sure," "Refuse," "Not asked or Missing". ItemIDs are used in the DCM processes as the representative of each (variable name: variable value). It is important to emphasize that a previously established assumption of DCM infers that an itemID can be considered as an

item or a granulized domain-concept. Variable names, descriptions, variable values, variable meanings, frequencies, and percentages are obtained directly from the provided BRFSS 2006 Codebook [33], which details all survey questions and their choices.

The *interesting indicators* in Table Appendix A.1 are implemented to differentiate itemIDs whether they are of an interest or not. The attributes with interesting indicator value N (no) usually imply that their variable meanings are not of health risks or problems [e.g. (*GENHLTH*: *1*), (*HLTHPLAN*: *1*)]. In addition, the interesting indicators are helpful in reducing the number of items to be considered in both the DCM and the brute-force frequent itemset mining of the "entire" set of transactions of the BRFSS 2006 data. The indicators exclude non-interesting itemIDs from a mining process's consideration. For example, the interesting indicators used in the domain-concept partition (*DIABETE2*: *1*) will report only results from items that imply health risks and problems. Therefore, itemID 1 will not be reported.

5.1 Computational Resources

There are two sets of the computation resources used in the experiments and evaluation processes. In the first experiment, the DCM approach was evaluated using a cluster system with 128 four-processor Intel Xeon CPU 2.66 GHz with four MB cache machines. These machines may contain four to six GB of memory. Please note that, there were two sets of 84 batch processes in the first experiment. In addition, at a given point in time, the processes may or may not utilize all of the 128 nodes of the cluster system.

The second experiment was conducted to evaluate the efficiency of the DCM-PA approach. For the purpose of monitoring the progress of the aggregation operations

executed by DCM-PA, the operations were evaluated using a single server with four-processor Intel Xeon CPU 2.80 GHz with a one MB cache and four GB of memory. Moreover, this server was also used as the storage device of the DCM offline results, where a MySQL database was utilized. Even though DCM-PA is designed to be used on-demand and on-line, all of the DCM-PA aggregation operations and their efficiency were tested as offline batch processes. This is to ensure that the experiment was not affected by the network and its traffic.

5.2 Offline DCM Frequent Itemsets

There are two main testing sets, which are directly related to the types of frequent itemsets used in the evaluation of DCM-PA. They are: 1.) DCM and DCM-PA based on the FPT algorithm [20] for brute-force frequent itemsets, and 2.) DCM and DCM-PA based on CHARM algorithm [49] for frequent *closed* itemsets. The testing for the brute-force frequent itemsets is to utilize the complete set of the frequent itemsets (and their subsets). The complete set is based on the downward closure property [1], which states that if an itemset is frequent then all of its subsets are frequent.

The testing for frequent *closed* itemsets is to obtain and use a lossless set of frequent itemsets (a set of frequent itemsets with minimum repeating subsets). Only the subsets of the frequent itemsets that are necessary for a complete transaction retracing and association rule generation purposes are kept. The set of frequent closed itemsets can be smaller than the set of frequent itemsets. Please note that the frequent closed itemsets are not same as the maximum frequent itemsets (the longest frequent itemsets). This is because one may not be able to retrace the actual numbers of transactions of the subsets

of the maximum frequent itemsets. The association rule generation cannot be done without the information of the number of transactions.

5.3 Experimental Results

The following section will outline and describe the statistical characteristics of the DCM and DCM-PA experimental results for both the brute-force frequent itemsets and frequent closed itemsets.

5.3.1 DCM Processes

Table 5.1 details the two sets of the DCM offline processes, where the first set was to mine the complete frequent itemsets, and the second set was to mine the frequent closed itemsets. A DCM process was implemented independently for each dc partition. The largest dc partition, (*CDVSTRK*: 2), has 341,643 transactions, which is 14,067 less transactions than the whole BRFSS 2006 data set (355,710 transactions in total). The smallest dc partition, (*EDUCA*: 1), had 630 transactions. For a purpose of comparing efficiencies based on sizes, the statistical power analysis was exempt from the pre-processing step. On average, a dc partition contains 85,738 transactions, with a standard deviation of 95,677 transactions. The global minimum support threshold used for all of the DCM processes was 0.1 (or 10%).

It is important to note that if we had mined all 355,170 transactions of the BRFSS 2006 at once with the same minimum support threshold of 0.1, we would not have uncovered any frequent (or frequent closed) itemsets from the domain-concepts that have less than 35,517 transactions. From Table Appendix A. 1, there are 46 domain-concepts

or variables (55% of 84 domain-concepts in total) with frequencies less than 35,517 transactions. More importantly, there are 39 domain-concepts (63% of 62 domain-concepts with the interesting indicator Y) that will not meet the threshold. This implies that the data mining results would likely have not contained findings that were related to health risks or problems. Hence, without DCM, the data mining effort and the results may have been rendered ineffective.

Table 5.1. Statistics of DCM Offline Processes for Frequent Itemsets and Frequent Closed Itemsets

Statistics	Number of Transactions (Domain-Concept Partition Size)	Brute-Force Time (seconds)	Frequent Closed Itemsets Time (seconds)
Maximum	341,643	84.4	1.2
Minimum	630	0.009	0.007
Average	85,738	8.9	0.3
Standard Deviation	95,677	17.6	0.3

Moreover, Table 5.1 details the statistics of the time spent (in seconds) during the offline processes. Overall, the time spent for the frequent closed itemset processes was shorter than for the frequent itemset processes. Figure 5.1 and Figure 5.2 show the corresponding dc partition size (number of transactions in $\mathbf{10^{-3}}$ unit) and time spent of each of the 84 dc partitions for the frequent itemsets (in seconds), and the frequent closed itemsets (in $\mathbf{10^{-2}}$ seconds), respectively. It is worth mentioning that the size of a dc partition may be the main factor, but it is not the only factor that contributes to the time spent for its offline DCM process. The other factors that may affect the time are the

distribution of the data (sparse or dense), and the load of the cluster server during testing. However, these other factors are not in the scope of this study.

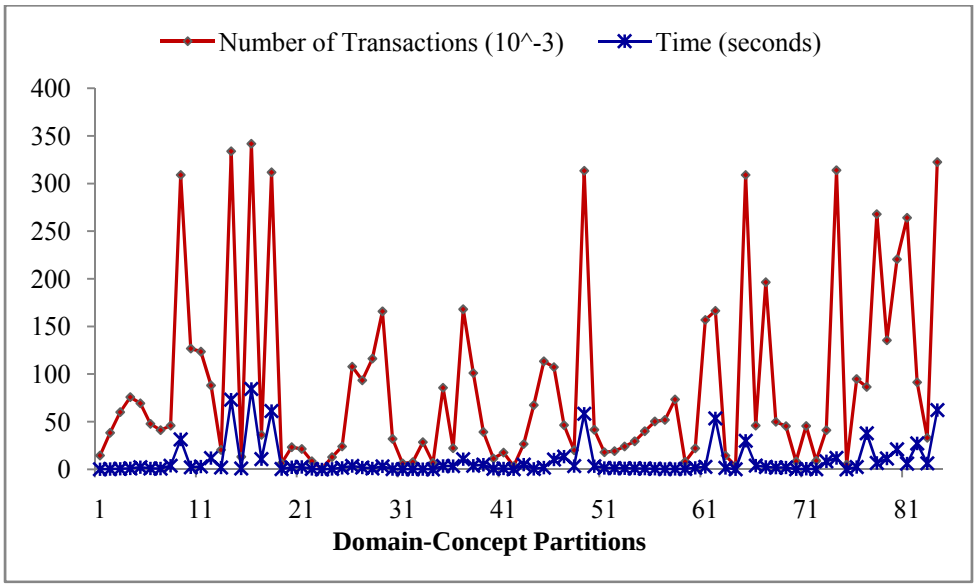


Figure 5.1. Numbers of transactions and time spent for the DCM offline frequent itemsets processes of the domain-concept partitions shown in Table Appendix A.1.

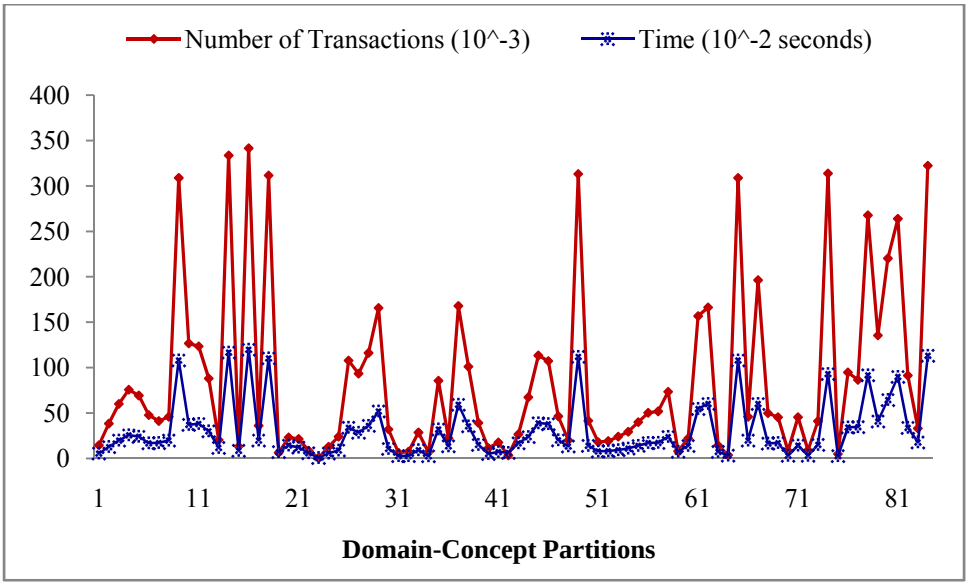


Figure 5.2. Numbers of transactions and time spent for the DCM offline frequent *closed* itemsets processes of the domain-concept partitions shown in Table Appendix A.1.

5.3.2 DCM-PA Processes

The experimental results are grouped according to the two approaches to aggregate dc partitions: 1.) union, which is to calculate $P(B|\vee A_i)$, and 2.) intersection, which is to calculate $P(B|\wedge A_i)$. The union operation for the brute-force frequent itemsets required at most 95,858 comparisons with the total time spent to merge all single frequent itemsets of 5.22 hours from 84 dc partitions. On the other hand, the union operation for the frequent closed itemsets required at most 84,344 comparisons with the total time spent of 5.19 hours from the same number of partitions. On average, the union operation between two dc partitions took about 0.2 seconds for each comparison regardless whether the comparison was for the frequent itemset or frequent close itemset. Further details of the union operation can be found in Table 5.2.

Table 5.2. Statistics of DCM-PA Processes Comparing between Aggregating Frequent Itemsets and Frequent Closed Itemsets Using the Union Operations

Statistics	Brute-Force Frequent Itemsets		Frequent Closed Itemsets	
	Number of Pair-Wise Comparison	Aggregation Time (seconds)	Number of Pair-Wise Comparison	Aggregation Time (seconds)
Maximum	2,923	634.8	2,686	657.8
Minimum	81	0.09	3	0.46
Average	1,183.4	232.2	1,041.3	230.9
Standard Deviation	734.2	163.5	799.1	191.5

The intersection operation of the brute-force frequent itemsets required at most 95,858 comparisons with the total time spent to aggregate all of the single frequent

itemsets of 7.8 hours from 84 dc partitions. On the other hand, the intersection operation for the frequent closed itemsets required at most 84,344 comparisons with the total time spent of 5.3 hours from the same number of partitions. On average, the intersection operation between two dc partitions also took about 0.2 seconds for each comparison regardless whether the comparison was for the frequent itemset or frequent close itemset. Further details can be found in Table 5.3.

Table 5.3. Statistics of DCM-PA Processes Comparing between Aggregating Frequent Itemsets and Frequent Closed Itemsets Using the Intersection Operations

Statistics	Brute-Force Frequent Itemsets		Frequent Closed Itemsets	
	Number of Pair-Wise Comparison	Aggregation Time (seconds)	Number of Pair-Wise Comparison	Aggregation Time (seconds)
Maximum	2,923	797.8	2,686	655.9
Minimum	81	32.9	3	0.8
Average	1,183.4	345.6	1,041.3	236.2
Standard Deviation	734.2	210	799.1	195.5

In the experiments of aggregating two dc partitions ($\mathrm{P}(B \mid A_1 * A_2)$, where * represents $\vee$ or $\wedge$). There were cases when frequent itemsets and frequent closed itemsets could not be aggregated due to: 1.) the first dc partition's frequent (and frequent closed) itemset was the same as the second dc partition (B is A_2), and 2.) the probability of $\mathrm{P}(B \mid A_1 * A_2) = 0$. Example of $\mathrm{P}(B \mid A_1 * A_2) = 0$ are

1.) B = (*TOTINDA*: *2*), A_1 = (*PROSTATE*: *2*), and A_2 = (*AGEG*: *1*),

2.) B = (*MARITAL*: *3*), A_1 = (*AGEG*: *1*), A_2 = (*QLACTLM*: *1*).

Please refer to Table Appendix A.1 for descriptions of the above examples. Table 5.4 concludes the number of no aggregation instances of both the brute-force frequent itemsets and the frequent closed itemsets.

Table 5.4. Number of "No-Aggregation" Instances Comparing between Brute-Force Frequent Itemsets and Frequent Closed Itemsets for Both Union and Intersection Operations

Case	Number of Instances	
	Brute-Force Frequent Itemsets	**Frequent Closed Itemsets**
B is A_2	134	112
$P(B \mid A_1 \vee A_2) = 0$	401	14
Total	535	126

5.3.3 Report of Correlation and Hybrid Values

As discussed in Section 3.3.2, the correlation coefficient (r), the coefficient of determination (r^2), and hybrid values (h) have been proposed. This section reports these values when DCM utilizes them to test for correlations between domain-concepts and items on the BRFSS 2006 data set. The average value of r's from the domain-concept partitions is 0.064. Furthermore, the average value of r^2's is 0.009. The low r and r^2 average values are contributed from the sparseness of the data and the number of different values that an attribute has. On average, an attribute has four values (min = 2 and max = 8). For example, r and r^2 values between the attributes *HLTHPLAN* and *GENHLTH* are 0.043 and -0.002, respectively. In this case, both *HLTHPLAN* and *GENHLTH* do not have a strong correlation (0.043), and the direction of the correlation is slightly negative. Moreover, hybrid values (h), which are calculated by equation (3.6) with the weight of support values (ω) as 2, are consequently low with an average of

0.001. More details of these r, r^2, and h, which are categorized by the attributes of domain-concepts, are shown in Table 5.5.

Table 5.5. Correlation Coefficient (r), Coefficient of Determination (r^2), and Hybrid Values (h) Categorized by Domain-Concepts' Attributes

Domain-Concept	Description	r	r^2	h
GENHLTH	General health	0.119	0.022	0.003
HLTHPLAN	Have any health care coverage	0.041	0.003	0.0002
DIABETE2	Ever told by a doctor you have diabetes	0.032	0.001	7.55E-05
CVDINFR3	Ever diagnosed with heart attack	0.058	0.004	0.0003
CVDSTRK3	Ever Diagnosed with A Stroke	0.047	0.003	0.0002
ASTHMA2	Ever told had an asthma	0.045	0.003	0.0002
QLACTLM2	Activity limitation due to health problem	0.136	0.031	0.0045
ORACE2	Respondent race choice	0.036	0.002	8.5E-05
MARITAL	Marital status	0.064	0.006	0.0005
EDUCA	Education level	0.174	0.038	0.0065
EMPLOY	Employment status	0.172	0.043	0.007
INCOME2	Income level	0.018	0.0004	1.19E-05
SEX	Respondents sex	0.015	0.000	6.02E-06
PROSTATE	Ever told you had a prostate cancer	0.029	0.001	3.92E-05
INSULIN	Now taking insulin	0.069	0.008	0.00067
DIABPILL	Now taking diabetes pills	0.015	0.0003	6.78E-06
FEETSORE	Ever had feet sores or irritations	0.035	0.0023	0.0001
DIABEYE	Ever told diabetes has affected eyes	0.035	0.003	0.0001
BMICAT	Computed Body Mass Index categories	0.077	0.007	0.0007

According to the characteristics of the BRFSS 2006 data set, the correlation and hybrid values are not sufficient to be used as independent thresholds because the reported

correlations among the variables are not strong. From this evaluation, the disadvantages of the under-represented groups (those with low support values, *s*), especially those (*attribute*: *value*) pairs with the interesting indicators 'Y', cannot be overcome by utilizing the correlation and hybrid values. This is because the hybrid values are calculated based on the harmonic mean calculation, which is considered as a measurement of *evenly* good performances or factors. In this case, the factors are *r*, r^2, and *s* values, which are unfortunately not high enough to yield good hybrid values.

5.4 Evaluation of 1-Itemsets

To evaluate the DCM approach, the experiments were conducted to compare: 1.) a distinct set of the 1-itemsets (itemsets of size one) uncovered from the 84 domain-concepts using the DCM approach, and 2.) the 1-itemsets uncovered from a frequent itemset mining using the FPT approach on the entire BRFSS 2006 data set. Table 5.6 contains the findings, which are the itemIDs with interesting indicator 'Y'. Empirically, the FPT approach implementation to the BRFSS 2006 with no domain-concept confirms that without domain-concept partitions, there can be many valuable findings that the brute-force approach cannot identify (see the cells in Table 5.6 with *). This is because these findings usually cannot meet the global minimum support threshold, such as the variables with health risks and problems with the interesting indicators 'Y', as discussed in section 5.3.1.

It is worth mentioning that there are two other major advantages from the DCM approach. First, a *x*-itemset from a dc partition implies a (*x*+1)-itemset, where *x* represents the number of items in the itemset, e.g. a co-occurrence of a 1-itemset and the

variable that the dc represents implies a 2-itemset. Second, an association rule "dc partition → *x*-itemset," where the support value of the *x*-itemset is the confidence value of the association rule.

Table 5.6. A Summary of the 1- Itemsets Results (ItemIDs) from Frequent Itemset Mining of the BRFSS 2006 Data Set (No Domain-Concept Partition)

ItemID from Table Appendix A.1	Results from Entire BRFSS 2006	
	ItemID (Uncovered*)	Support Value
4	4	0.13
5	n/a*	n/a
7	7	0.117
9	9	0.882
10	10	0.101
12	n/a*	n/a
14	n/a*	n/a
16	16	0.129
18	18	0.243
20	n/a*	n/a
22	22	0.552
23	23	0.14
24	24	0.127
26	26	0.128
27	n/a*	n/a
29	n/a*	n/a
30	n/a*	n/a
31	31	0.303
32	32	0.263
33	33	0.327
34	34	0.466
35	n/a*	n/a
38	n/a*	n/a
39	n/a*	n/a
40	40	0.241
41	n/a*	n/a
42	n/a*	n/a
43	n/a*	n/a

ItemID from Table Appendix A.1	Results from Entire BRFSS 2006	
	ItemID (Uncovered*)	Support Value
44	n/a*	n/a
45	n/a*	n/a
46	46	0.112
47	47	0.141
48	48	0.146
49	49	0.206
50	50	0.381
51	51	0.619
56	56	0.11
57	n/a*	n/a
58	n/a*	n/a
61	n/a*	n/a
63	n/a*	n/a
65	n/a*	n/a
67	n/a*	n/a
69	n/a*	n/a
71	n/a*	n/a
72	72	0.108
73	73	0.169
74	74	0.213
75	75	0.195
76	76	0.134
77	77	0.116
79	79	0.257
81	81	0.129
82	82	0.356
83	83	0.347

A summary of the 1-itemsets, their minimum and maximum support values, and the number of dc partitions that the 1-itemsets are associated to is shown in Table 5.7. The shaded rows are corresponding to the uncovered frequent itemsets from the brute-force approach as previously shown in Table 5.6.

Table 5.7. A Summary of the DCM 1-itemsets (ItemIDs), Their Minimum and Maximum Support Values, and the Number of Domain-Concept Partitions

ItemID	Min Support Value	Max Support Value	Number of DC Partitions
4	0.101	0.342	66
5	0.103	0.401	33
7	0.101	0.428	56
9	0.572	0.97	82
10	0.1	1	49
12	0.104	0.293	27
14	0.101	0.189	10
16	0.102	1	78
18	0.104	0.815	81
20	0.102	0.465	43
22	0.161	0.82	78
23	0.103	0.286	71
24	0.107	0.543	57
26	0.101	0.682	54
27	0.1	0.1	1
29	0.104	0.144	6
30	0.101	0.197	33
31	0.134	0.429	78
32	0.196	0.458	78
33	0.105	0.634	76
34	0.103	0.682	75
35	0.101	0.164	13
38	0.1	0.171	9
39	0.25	0.25	1
40	0.104	0.799	68
41	0.102	0.414	34
42	0.1	0.245	24
43	0.101	0.176	22

ItemID	Min Support Value	Max Support Value	Number of DC Partitions
44	0.1	0.146	28
45	0.1	0.147	38
46	0.101	0.146	62
47	0.102	0.169	64
48	0.102	0.198	49
49	0.1	0.4	45
50	0.173	1	81
51	0.45	0.988	80
56	0.1	0.352	63
57	0.202	0.266	2
58	0.101	0.256	11
61	0.109	0.258	10
63	0.11	0.478	8
65	0.104	0.787	23
67	0.101	0.211	4
69	0.15	0.395	7
71	0.102	0.529	5
72	0.101	0.275	43
73	0.102	0.25	60
74	0.114	0.302	73
75	0.141	0.355	74
76	0.106	0.372	62
77	0.102	0.493	46
79	0.122	0.605	82
81	0.102	1	78
82	0.132	0.55	81
83	0.239	0.476	81

There are 2,564 1-itemsets (itemsets of size 1) in total that are uncovered from the 84 dc partitions, with an average of 31 1-itemsets per partition. The itemIDs that cannot be uncovered from the mining of the entire set of transactions from BRFSS 2006 data are in shaded areas. Please note that the uncovered items contribute to 23 of 55 interesting itemIDs. If one focuses only on the shaded areas of both tables, one would find that DCM with dc partitions can report more associations than without dc partition ranging from an association with 1 dc to 43 dc's. Hence, it confirms that, without dc partition, there can be interesting items and their associations with one or more dc's missing from the results.

5.5 Evaluation of Itemsets with Other Sizes

Table 5.8 summarizes the brute-force frequent itemset mining on the entire set of transactions from the BRFSS 2006 data. The longest frequent itemset uncovered with the minimum support threshold of 0.1 is 4.

Table 5.8. A Summary of the Results from the Brute-Force Frequent Itemset Mining on the Entire BRFSS 2006 Data Set

Itemset Size	Number of Itemsets	Minimum Support Value	Maximum Support Value
1	32	0.101	0.882
2	74	0.538	0.103
3	51	0.283	0.102
4	9	0.137	0.101

Table 5.9 details the nine 4-itemsets. When each of the itemIDs is interpreted using Table Appendix A.1, one may find that the nine of these 4-itemsets contain mostly demographic information, and very little information regarded health risks and problems.

Table 5.9. Nine 4-Itemsets Uncovered Using the Brute-Force Frequent Itemset Mining on the Entire BRFSS 2006 Data Set

Itemset No.	ItemID	ItemID	ItemID	ItemID	Support Value
1	22	51	34	9	0.137
2	51	22	82	9	0.121
3	22	34	50	9	0.117
4	22	34	33	9	0.115
5	22	34	49	9	0.107
6	22	51	33	9	0.106
7	22	33	49	9	0.105
8	22	50	83	9	0.103
9	51	34	33	9	0.101

In contrast, there are five instances of 6-itemsets uncovered from the DCM approach as shown in Table 5.10. Again, a 6-itemset uncovered from a dc partition implies a 7-itemset. Further, from the DCM approach, there are 378 instances of 5-itemset, 2,575 instances of 4-itemset, 6,893 instances of 3-itemset, and 7,588 instances of 2-itemset in total. Also important is the fact that the association mining of a data set without domain-concept partitions yields unorganized associations. Hence, drawing a conclusion from the findings would take more effort. In conclusion, there are a big numbers of associations that a brute-force mining approach could not uncover without partitioning the data.

Table 5.10. Six Itemsets Uncovered Using the DCM Approach from the BRFSS 2006 Data Set with Domain-Concept Partitions

Itemset No.	DC Partition	ItemID	ItemID	ItemID	ItemID	ItemID	ItemID	Support Value
1	52	9	22	40	50	77	83	0.106
2	67	9	10	18	20	51	79	0.105
3	67	9	10	18	20	65	79	0.105
4	67	9	10	18	51	65	79	0.101
5	67	9	10	18	20	63	79	0.1

5.6 Evaluation of DCM-PA Itemsets

In order to demonstrate that the DCM-PA approach can compliment DCM to uncover many more itemsets than the brute-force frequent itemset mining approach, an example from Table 5.6 is used. Let us investigate the 1-itemset with the itemID 10, which is (*DIABETE2*: 1) or "*Have you ever been told by a doctor that you have diabetes?*: *yes*," with the support value of 0.101. There are no co-occurrences of the itemID 10 with other itemIDs uncovered from the BRFSS 2006 without dc partition. In other words, the itemID 10 is not a part of any other size itemsets; hence, it has no association uncovered.

On the other hand, there are 49 dc partitions where the DCM approach uncovered the itemID 10 as their 1-itemset, with the support values ranging from 0.1 to 1 (see Table 5.7 at itemID 10). More importantly, using the DCM-PA approach to aggregate dc partitions in the form of $P(B \mid A_1 \vee A_2)$, where A_1 is the itemID 10, A_2 can be the other 83 dc partitions ($A_1 \neq A_2$), and B can be the other 82 itemIDs ($B \neq A_1 \neq A_2$), there are 1,828 such $P(B \mid A_1 \vee A_2)$ instances as the results of the DCM-PA union operations. The summary of the results can be found in the Appendix B. Moreover, if one wishes to aggregate A_1 with multiple other domain-concepts in order to further expand its associations, one can do this freely through an online system, DCMiner, which all offer $P(B \mid A_1 \wedge A_2)$ as well.

One way to validate the DCM-PA approach is to demonstrate that the number of transactions that made up $\left(B \wedge (\wedge A_i)\right)$, which can be derived from the instance

$P(B \mid \wedge A_i)$ (see equations (4.18) to (4.20) in Chapter 4), is the same as the result of a multiplication between:

1.) the support value of the brute-force approach's itemset (with the size of three or larger when DCM-PA aggregates two domain-concept partitions, or $P(B \mid A_1 \wedge A_2)$). Itemsets are those that have B and A_i as its items, and

2.) the total number of transactions in the BRFSS 2006 data set.

In the case of three variables (B, A_1, and A_2), without the loss of generalization, the support values and the numbers of transactions of B, A_1, $A_2, B \wedge A_1$, $B \wedge A_2$, and $A_1 \wedge A_2$ that made up the value of $P(B \mid A_1 \wedge A_2)$, were calculated by using the exact same algorithm for both the DCM approach and the brute-force frequent itemset mining on the BRFSS 2006 data set without domain-concept partition. Therefore, it is adequate to draw a conclusion that a DCM-PA's $P(B \mid A_1 \wedge A_2)$ value is correct once compared to the support value of the brute-force frequent 3-itemset of the same variables (B, A_1, and A_2).

CHAPTER 6

APPLICATIONS

In the recent years, Domain-Concept Mining (DCM), DCM Partition Aggregation (DCM-PA), and DCM Web system, called DCMiner, have been successfully implemented as data mining tools for various data sets, including: 1.) the *Agency for Healthcare and Research Quality* (AHRQ)'s *Nationwide Inpatient Sample* (NIS) data [35], 2.) the *Center for Disease Control and Prevention* (CDC)'s *Behavioral Risk Factor Surveillance System* (BRFSS) data [36], 3.) the data sets from the *National Institute of Health (NIH)-funded breast cancer survivors with lymphedema project*, and 4.) an industrial engineering grouping technology data. Other data sets where DCM and DCMiner have been implemented, but are omitted from the detailed discussion includes the *Callaway nuclear power plant's Action Request* (CAR) data and geospatial image databases [92]. A summary of the implementations categorized by the applications is shown in Table 6.1.

Table 6.1. A Summary of DCM, DCM-PA, and DCMiner Applications

Data Set	DCM	DCM-PA	DCMiner	Future Work
NIS [29]	Multi-level domain-concepts	On-going	Yes, with both tabular and graphical formats.	Temporal (years prior to and after 2005) and spatial domain-concepts for trends
				DCM-PA
				DCMiner with Google Earth [93]
BRFSS [94]	Yes for 2003-2006 data sets	Yes, 2006 only	Yes, with tabular format for all domain-concepts, but graphical format for (*diabetes*: *yes*) only.	DCM-PA implementations for other data years
				More domain-concepts, include states (spatial) and non-core questions
				Full DCMiner graphical functionalities
				Expand to cover 1990 – 2002 data sets
Breast cancer survivors with lymphedema [95]	Yes, but limited to cancer-affected sides, limb dominant sides, and BMI categories as domain-concepts	No	Yes with both tabular and graphical format, but limited functionalities.	More domain-concepts include medication history, breast cancer treatments, signs and symptoms, among others.
				Expansion of DCM to uncover over-time trend of limb volume change and its associations with other risk factors
				Full DCMiner functionality with DCM-PA
Industrial engineering grouping technology [96]	Yes, with Sequential Forward Floating Selection	No	No. Results are off-line machine/cost (m/c) matrices only	n/a
CAR	Yes, with text mining	No	Yes with tabular format only.	Expansion of DCM to be able to rank and classify the results, i.e. to generate 'association rule induction' and over-time patterns from the uncovered associations
Geospatial image databases [92]	Yes, for ranking and classification purposes	No	No. Results are off-line features of images	n/a

DCM and its related approaches have been useful to a vast range of data sets because DCM can uncover valuable associations among attributes from these rich data sources efficiently; whereas, the traditional association mining may not be efficiently or feasibly applied. Further, the findings from DCM are highly organized according to a data set's domain-concept (dc) partitions. In this chapter, some of the research that havs implemented DCM, DCM-PA, and DCMiner are discussed.

6.1 DCMiner

DCMiner was originally designed by a team of three doctoral students to create a back-end relational database for storage of DCM results, and to create an interface which offered various visualizing formats for the results. This work was named as a winner of the *American Medical Informatics Association* (AMIA) 2007 Annual Symposium Data Mining Competition [29].

Moreover, DCMiner offers more functions than Web-based database browser of the DCM mining results. DCMiner also offers a tool through which human experts can view, compare, and contrast the results across domain-concepts in both tabular (text) and graphical formats. In addition, DCMiner has the ability to present the findings on-the-fly in graphical formats, particularly when there are thousands of potentially interesting findings to browse. A graphical representation can act as an informative summary of a group of findings. Consequently, the number of findings presented to the human experts is reduced.

In general, findings that can be identified with trends or patterns across some certain sets of domain-concepts, including temporal (e.g. years), spatial (e.g. states), and

other domain-concepts (e.g. age groups, income levels, races, genders, levels of healthcare coverage, etc.), DCM's representations also allow human experts to contemplate a more complete picture of the findings, rather than just a snapshot (a table) would be able to provide. More details of the DCMiner implementations for the NIS, BRFSS, and lymphedema data are discussed as part of sections 6.2, 6.3, and 6.4, accordingly.

6.2 Domain-Concept Mining on the 2005 Nationwide Inpatient Sample (NIS) Data

The 2005 Nationwide Inpatient Sample (NIS) contains almost 8 million transactions. Datasets of this size can be under-utilized due to their complexity and the difficulty in comprehending and exploring the relationships among variables. To exploit this rich data set, we applied DCM and discovered approximately 8.9 million frequent itemsets in our 149 partitions of the 2005 NIS. DCMiner was then used to facilitate the navigation of the results, and the research community was able to identify clinically meaningful patterns in the NIS dataset for further examination and analysis. Thus, the DCMiner demonstrates the potential for using computational methods to provide an efficient, robust, and flexible tool to healthcare researchers for knowledge discovery, which may lead to further clinical studies.

6.2.1 Motivations

The 2005 NIS is a collection of inpatient visits from hospitals in 37 states. The hospitals included in the study were chosen using stratified sampling, with strata

definitions based on the hospital characteristics of geographic region, control, location, teaching status, and bed capacity. Between these five characteristics, a total of 14 strata were defined, with sampling performed to select 20% of the all information in each stratum. From the 3,860 hospitals obtained using this sampling strategy, nearly 8 million records for the dataset were selected.

Despite the fact that the 2005 NIS is a sampling itself, the dataset is quite large, and the ability to utilize such large datasets for knowledge discovery is important, as these types of datasets are becoming increasingly more available [97].

Some previous studies utilizing the NIS dataset performed statistical analysis on limited topics from the dataset. These types of studies are valuable in the topic of interest, but are incapable of discovering associations across or among other topics. Data mining approaches, however, facilitate the discovery of such associations. Unfortunately, with a traditional, purely descriptive data mining approach, such as association rule (AR) mining [13], the knowledge discovered is unorganized and the sheer volume of extracted information is overwhelming to researchers. It therefore becomes essential to develop methods of organizing the discovered knowledge and presenting it in meaningful ways so that newly discovered knowledge can be identified by humans [98].

Domain-Concept Mining (DCM) is a customized, semi-descriptive data mining approach that seeks to accomplish these goals. DCM first organizes data before analysis by partitioning the data into groups of relevant domain-concepts. For the 2005 NIS, these domain-concepts are based mainly on Clinical Classifications Software (CSS) for ICD-9-CM as well as demographic, temporal, and spatial variables. The DCM then analyzes data by extracting associations among variables from each partition. This is accomplished

in a two-step process: 1.) off-line discovery of associations from each domain-concept partition, and 2.) on-line exploration of potential relationships between selected domain-concepts and other variables of interest via a web application called DCMiner. This tool allows researchers to efficiently navigate the findings from each domain-concept, to compare and contrast findings across domain-concepts, and to view results using a variety of visualization techniques.

6.2.2 Knowledge Discovery Process

The process by which one extracts, uncovers, and identifies meaningful and relevant knowledge in large datasets is known as the knowledge discovery process. This process is comprised of four steps: 1.) data selection, 2.) data pre-processing and transformation, 3.) data mining, and 4.) presentation of results. We discuss each step of the process with respect to our analysis of the 2005 NIS.

6.2.2.1 Data Selection

To increase the efficiency of the mining process, DCM implements a stratified random sampling technique, which not only reduces the number of transactions, but maintains the characteristics of the original data. According to statistical sample size estimation [31], a dataset containing 8 million records can be represented with as few as 9,604 random samples with a confidence interval 95%. Considering computational resources, the time constraints for analysis, and the fact that the NIS dataset is already a sampled dataset, we chose to perform stratified sampling (based on the CDC's predefined age groups [33]) to select 10% of the data, which is a much more conservative sampling than theory requires. To ensure accuracy, we conducted an experiment to compare

attribute value (item) frequencies between the raw NIS dataset and our sampled set. In terms of total number of items occurring with a frequency of 10% or more, the sampled and NIS data matched 358 of 359 items. Furthermore, the frequency values of these items differed by at most 0.5%.

6.2.2.2 Data Pre-processing and Transformation

As with any dataset, a certain amount of pre-processing had to be performed on the 2005 NIS before data analysis could be initiated. First, the degree of the dataset was reduced to eliminate redundancies and sparsely populated attributes. In total, 60 attributes were selected for study. These attributes include information related to demographics, hospital, diagnoses, comorbidities, procedures, admission types, among others. Second, because descriptive data mining techniques rely on co-occurrence frequencies, the attributes of interest need to be discrete in nature. Thus, discretization of attributes with continuous values must be performed before the data can be analyzed with DCM. Some continuous attributes that were discretized include age, total charge, length of stay, number of procedures, and the number of days from admission to principal procedure. For the age attribute, discretization was performed as mentioned previously using CDC guidelines. For the other continuous attributes, statistics (e.g. range, mean, mode, and standard deviation) were used as the underlying guides in defining partitions.

6.2.2.3 Data Mining: Domain-Concept Mining

Descriptive data mining approaches aim to uncover hidden knowledge by identifying itemsets (a set of co-occurring items) without prior hypotheses. For example, the (*attribute*: *value)* pairs {(*GENDER*: *Female*), (*PRIMARY PAYER*: *Private including*

HMO), (*MDC*: *Pregnancy, Childbirth, and Puerperium*)} together form an itemset. Since the approach was developed [13], researchers have struggled with one of its challenges: how to discover novel, useful, or interesting frequent itemsets efficiently. This is difficult because on the one hand, itemsets with high support (co-occurrence frequency) are usually trivial knowledge, but on the other hand, itemsets with low support are often novel and clinically significant. Because of this, setting the support threshold is vitally important. If the support threshold is set too high, important associations of rare itemsets will never be discovered. If the threshold is set too low, analysis becomes computationally inefficient and even impracticable due to the huge number of qualified frequent itemsets.

For example, consider the itemset (*RACE*: *Native American*), which has a probability of 0.0033 in the 2005 NIS. If one would like to find novel knowledge from this sub-population, the support threshold should be set lower than 0.33%. With such a low threshold, millions of qualified co-occurring itemsets will make it difficult for users to find the proverbial needles in the haystack of returned itemsets to identify meaningful hidden knowledge.

To address this issue, we define domain-concepts by partitioning the 2005 NIS dataset into two levels of domain-concepts. First-level domain-concepts can be grouped into demographic, temporal, spatial, comorbidity measures (CM), and major diagnosis category (MDC) attributes, among others. All first-level domain-concepts are listed in Table 6.2, along with the numbers of their level-two concepts. For example, the first-level domain-concept "Hospital Bed Size" has three level-two concepts: "Small," "Medium," and "Large."

Table 6.2. Lists of the Number of Level-Two Domain-Concepts under Each First-Level Domain-Concept

Level-One Domain-Concepts	# of Level-Two Domain-Concepts
Hospital State	37
Comorbidity Measures	29
MDC	25
Admission Month	12
Age Groups	8
Age Groups (100%)	8
Race	6
Control of Hospital	5
Discharge Quarter	4
Hospital Region	4
Bed Size of Hospital	3
Survival	2
Gender	2
Hospital Location	2
Hospital Teaching Status	2

Let X be a subset of items (variables) in database, *dx* be X's domain-concept, and T be the set of all records in the database. A subset of T that shares the same domain-concept *dx* is defined as: $T_{dx} = \sigma_{dx}(T)$, where σ is the relational algebra selection operator. With domain-concept partitioning, one record could be in multiple partitions since the record may be qualified for many domain-concepts according to its attribute values. Records in T_{dx} will then be mined to extract frequent itemsets. An itemset is said to be frequent in a domain-concept *dx* if and only if $\frac{|T_{dx}^I|}{|T_{dx}|} \geq s$, where T_{dx}^I is the set of

records in *dx* that contain *I*, *s* is the minimum support threshold, and |.| is the number of records from a dataset.

While one of the advantages of data mining is that the results are driven by the data and not constrained by preconceived hypotheses, the use of domain-concepts is useful in providing a seed or starting point for uncovering knowledge. The search for other interesting, relevant, and nontrivial information can be extended from the original domain-concepts.

Offline processing was performed on a Dell EM64T cluster system with 128 nodes and 512 processors. Each node has 4-6 GB of memory and is attached to 50 TB of disk using an Infiniband high speed interconnect infrastructure for both processor and storage access.

6.2.2.4 Knowledge Presentation and Visualization

We designed a web-based, database-driven knowledge presentation and visualization system called DCMiner that assists users in navigating the information uncovered from DCM. This website has two main functional components: 1.) a text-based, result-browsing capability (Figure 6.1), and 2.) a graphical representation of statistical distributions (Figure 6.2).

In browse mode, users must first initialize the knowledge interpretation process by choosing a number of criteria to select a subset of the result set as shown in the top panel of Figure 6.1. For this, users are prompted to select a level-two domain-concept, chosen via two levels of dropdown menus that correspond to the two levels of domain-concepts. Other criteria provided to the user are the maximum number of co-occurring

items, a minimum frequency threshold, and a list of attributes of interest from which the user can select multiple items.

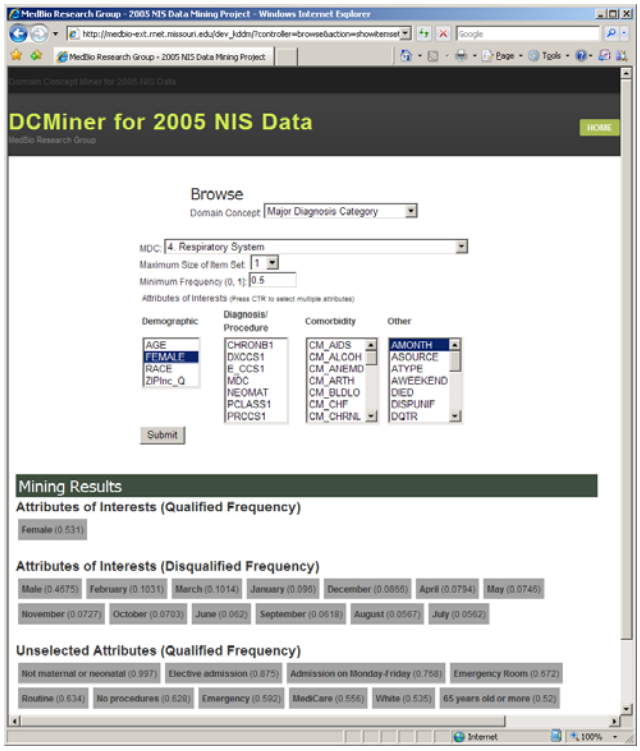


Figure 6.1. A Screenshot of DCMiner.

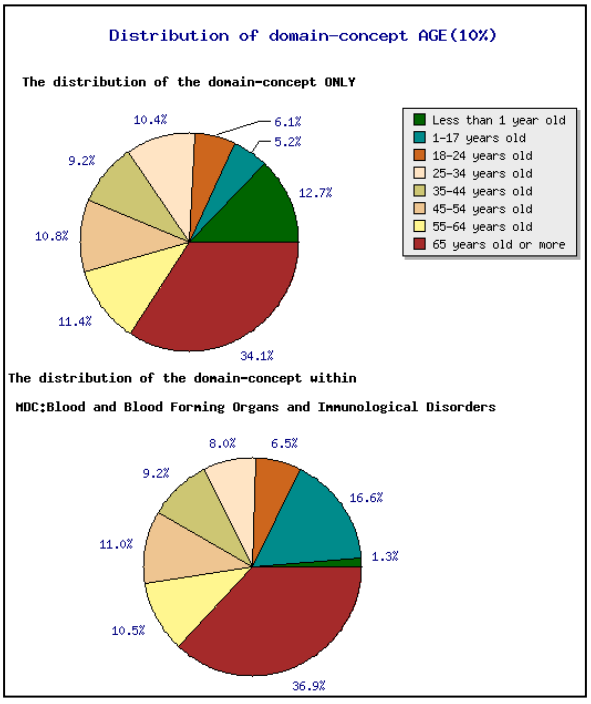


Figure 6.2. Graphical Representation of Statistical Distribution.

The lower panel of Figure 6.1 contains: 1.) a list of qualified, selected 1-itemsets ranked by their co-occurrence frequencies within the domain of interest, 2.) an expanded list of qualified 1-itemsets that were not selected but might be relevant, and 3.) a list of disqualified, selected 1-itemsets that did not pass the frequency threshold. The users are able to check a set of 1-itemsets from 1.) and 2.) for further studies.

Those qualified 1-itemsets that were not checked along with the disqualified 1-itemsets will not be used in forming the larger itemsets, which range in size from two to the selected maximum number of co-occurring items. Larger itemsets (not shown due to space limitation) are then summarized in ranked order based on their co-occurring frequencies.

For example, consider the situation in Figure 6.3, where "Race" was chosen at the level-one domain-concept, and "Black" the level-two domain-concept. Users can choose one of the qualified 1-itemsets such as "Emergency" to view a comparison of this itemset across all level-two sub-domain siblings. Users can also view larger itemsets containing those qualified 1-itemsets.

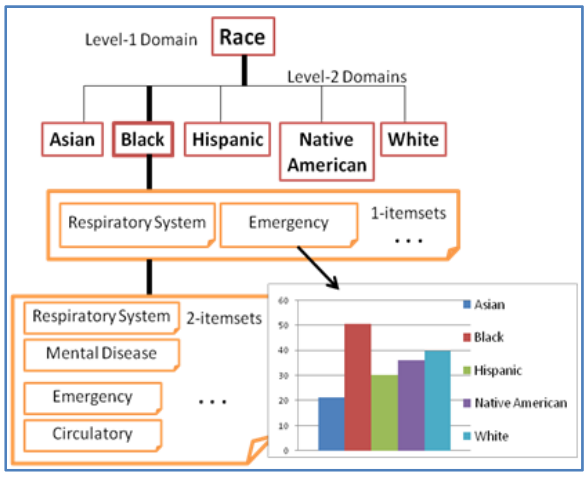


Figure 6.3. Compare and contrast specific itemsets among level-two sub-domain siblings.

In the graphical mode, charts are generated on-the-fly based on selected domain-concepts. An example is shown in Figure 6.2, where the top pie chart depicts the distributions of the age group domain-concepts while the bottom pie chart shows the distribution of the same domain-concept within the selected variable (*MDC*: *Blood, blood-forming organs, and immunological disorders*) for comparison purposes. These two charts appear side-by-side and can be used to learn differences the selected variable have on the domain-concept's distribution.

The web-based online system is a PowerEdge 1850 with Dual Xeon-4, 2.8 GHz processors and 4 GB of memory.

6.2.3 Results

After data mining has completed, DCM has identified all frequent itemsets in the dataset whose co-occurrence frequencies are greater than the support threshold. In total, this is nearly 8.9 million itemsets for the 2005 NIS, any of which may potentially qualify as a meaningful result. Due to space limitations, we report only a select subset of those findings in this section. The reported findings are classified into four broad classes of domain-concepts: demographic, temporal, spatial, and diagnosis attributes. Additional results can be generated and examined using the DCMiner at http://medbio-ext.rnet.missouri.edu/kddm.

6.2.3.1 Demographic Attributes

Findings in this broad domain-concept class are based on demographic information like gender, race, income level, and age. Example findings include: 52.2% of Caucasian admissions live in large metropolitan areas versus 72.6% of Asians admissions

and 30.8% of Native American admissions; 4.52% of males have the CM of depression compared to 6.91% of females; and 31.76% of people living in zip codes with a median household income of less than $37,000 have the comorbidity measure of alcohol abuse versus only 16.85% of those living in zip codes with a median household income greater than $61,000.

6.2.3.2 Spatial Attributes

Findings in the spatial domain-concept class are in some way linked to the location and characteristics of hospitals. For example, the most frequent MDC for the Midwest, Northeast, and South was contained diseases and disorders of the circulatory system, but was contained in pregnancy, childbirth, and puerperium for the West.

6.2.3.3 Temporal Attributes

Findings in the temporal domain-concept class are related to attributes like admission month and discharge quarter. An example finding would be that admissions with an MDC of diseases and disorders of the respiratory system occur more frequently in winter months (e.g. 10.3% in February and 9.6% in January) than in summer months (e.g. 5.6% in July and 5.7% in August).

6.2.3.4 Diagnostic Attributes

Among those with a CM of obesity, 53.2% are from large metropolitan areas, 27.6% from small metropolitan areas, and 11.5% from micropolitan. The gender breakdown for this subpopulation is 64.6% female compared to 35.4% male. The top three ranked MDC in this group are hypertension (58.9%), diabetes (32.6%), and chronic

pulmonary disease (22.6%). More details regarding the approximate 500 ICD9-CM CCS for diagnoses and procedures can be viewed for this population using DCMiner.

Another example population consists of those with a CM of metastatic cancer. In this group, 53.2% are female versus 46.8% male; and 53.5% are from large metropolitan areas, 27.5% from small metropolitan areas, and 10.5% from micropolitan areas. The top three ranked MDC for this group are hypertension (36.9%), fluid and electrolyte disorders (28.1%), and chronic pulmonary disease (19.8%). Other relevant associations include 48.3% use the ER, 72.3% chose elective admissions, 55.9% were paid by Medicare, and 37% were discharged within 5-10 days.

6.2.4 Conclusion

In this competition, the proposed semi-descriptive data mining approach offers unlimited possibilities for discovering and exploring new knowledge buried in the 2005 NIS via efficient mining using DCM and a web-based visualization tool.

6.3 Knowledge Discovery using Domain-Concept Mining Approach for the Behavioral Risk Factor Surveillance System (BRFSS) Data

The publicly available, state-based *Behavioral Risk Factor Surveillance System* (BRFSS) data from the Centers for Disease Control and Prevention (CDC) is the largest annual state-based telephone health survey system in the world. Often times, the data set is under-utilized due to its size, complexity, inconsistency of variables used among the survey years, and the difficulty of comprehending and exploring the relationships among variables. With a traditional data mining approach, such as AR mining, the amount of

knowledge discovered is too overwhelming for policy makers to efficiently manage. The DCM approach partitions data into groups of relevant domain-concepts, mainly health-related and demographic variables, then extracts associations among other variables from each partition. The DCM is a two-step process: 1) off-line discovery of associations from each domain-concept partition and 2) on-line exploration of potential relationships between selected domain-concepts and other variables of interest.

The DCM on-line system provides public health policy makers with a tool to investigate potential health related issues in a variety of dimensions across various risk factors. These factors include geographic location, times, availability and access to health care providers and health insurance, education, income, race, age, disease of interest (i.e., diabetes or HIV), weight and body mass index levels, exercise regimens, nutrition, smoking, etc. For example, an association rule from this health domain-concept could possibly be rendered in questions (and a calculated value of a question), such as

> "{(*Have you ever been told by a doctor that you have diabetes?*: *yes*)} → {(*Have you ever had blood cholesterol checked?*: *yes*) **AND** (*At risk for heavy alcohol consumption (greater than two drinks per day for men and greater than one drink per day for women)*: *not at risk for heavy drinking*)}."

Similar rules for the same domain-concept (*diabetes*) are also extracted and archived in a relational database for knowledge aggregation.

These domain-concepts are used to assist in policy planning, in which the DCM finding results can help pin-point health related problems for further examinations and analyses for specific populations or for generalized uses. Moreover, a portion of the findings from the BRFSS data sets between 1990 and 2006 show that DCM may

efficiently discover valid and relevant information from the BRFSS with respect to previously published literature, such as those in *PubMed*. In addition, findings from DCM may also potentially suggest further studies of the new knowledge that has not been published.

6.3.1 Motivations

- To discover hidden knowledge in large databases, such as a publicly available data set from the Centers for Disease Control and Prevention (CDC)'s Behavioral Risk Factor Surveillance System (BRFSS) questionnaire [36]. A BRFSS data set can contain more than 1,000 items (from 302 attributes in BRFSS 2006, each of which has approximately 5 different answers or values), and 181, 289 records yearly average (maximum 355,710 records in 2006).
- From this data set size, the traditional association rule mining algorithm, such as the Apriori algorithm [13], could take days to discover, and it would result in millions of association rules. In most threshold values setting, it is not feasible to discover association rules directly from the entire data set.
- Not all discovered association rules are relevant, useful, and novel. Not all items should be included in the data mining process.
- BRFSS items can be categorized into domain-concepts. The example of domain-concepts are health-related behavior, health-related risk, nutrition intake, exercises, etc.
- A domain-concept is a set of items of interest. The domain-concept helps categorize and identify relevant items. We can separately implement data mining

process on each domain-concept. This can make the data mining process faster and more feasible.

6.3.2 Methods

In BRFSS data, a domain-concept of interest is a (*attribute*: *value*) pair, for example (*Do you have any health care coverage?*: *No*). Let X be a subset of items (*attribute*: *value*) in database R, dx be X's domain-concept, and T be a set of all records in R. A subset of T that shares the same domain-concept dx is defined as:

$$T_{dx} = \sigma_{dx}(R) \tag{6.1}$$

, where σ is a selection operator. Records in T_{dx} will then be mined to extract frequent itemsets.

An itemset I is a set of (*attribute*: *value*) pairs. It is said to be frequent in a domain-concept if and only if

$$\frac{|T_{dx}^{I}|}{|T_{dx}|} \geq s \tag{6.2}$$

, where T_{dx}^{I} is the records in dx that has I, s is the minimum support threshold, and |.| represents the number of records from a data set. Frequent itemsets are extracted only from partitions with sizes that are smaller than or equal to

$$\frac{|T|}{N_{dx}} \pm \varepsilon \quad | \quad \varepsilon \geq 0 \tag{6.3}$$

, where N_{dx} is the total number of domain-concepts. It gives us the expected number of records for each domain-concept that we will conduct experiments on. If a partition is too large, it suggests to further divide itself into multiple domain-concepts.

6.3.3 Example of Findings

Selected findings and associated support values as shown in Table 6.3, Table 6.4, and Table 6.5 are from BRFSS 2003 to 2006 data sets. Their co-occurring support values are 23.4%, 24.7%, and 24.2%, respectively. The findings are from the domain-concept: (*Ever been told by a doctor that you have diabetes*: *yes*), with criteria of co-occur support values ≥ 20.0%.

Table 6.3. Findings from BRFSS 2003 with a Co-Occur Support Value 23.4 %

Findings	Support Values (%)
(*Fruits or vegetables consumption per day*: *consume < 5 servings per day*)	74.2
(*Ever been told blood pressure high*: *yes*)	68.0
(*Trying to lose weight*: *yes*)	61.1
(*Body Mass Index*: *obese (30.0 ≤ BMI < 99.99)*)	59.3

Table 6.4. Findings from BRFSS 2004 with a Co-Occur Support Value 24.7 %

Findings	Support Values (%)
(*Race groups*: *white (non-hispanic)*)	71.7
(*Body Mass Index*: *obese (30.0 ≤ BMI < 99.99)*)	59.6
(*Respondents sex*: *female*)	57.3

Table 6.5. Findings from BRFSS 2005 with a Co-Occur Support Value 24.2 %

Findings	Support Values (%)
(*Fruits or vegetables consumption per day*: *consume < 5 servings per day*)	73.7
(*Race groups*: *white*)	72.0
(*Ever been told blood pressure high*: *yes*)	69.4
(Ever been told blood cholesterol high: *yes*)	59.2

Table 6.6. Findings from BRFSS 2006 with a Co-Occur Support Value 20.2%

Findings	Support Values (%)
(*Was there a time in the past 12 months when you needed to see a doctor but could not because of cost*: *no*)	86.7
(*Are you now taking diabetes pills*: *yes*)	59.2
(*Physical activity of exercise in the last 30 days*: *no*)	40.1

The number of surveys increases each year as shown in Table 6.7. It is noteworthy to mention that the percentage of the selected domain-concept, (*Ever been told by a doctor you have diabetes*: *yes*) also increases each year.

Table 6.7. The Number of Surveys from the Selected Domain-Concept Comparing to the Entire BRFSS 2003 - 2006 Data Sets

Years	Number of Surveys (records)	Number of Surveys from the Selected Domain-Concept
2003	264,684	21,729 (8.2%)
2004	303,822	25,736 (8.5%)
2005	356,112	33,320 (9.4%)
2006	355, 170	36,085 (10.1%)

Table 6.8 shows the support values of various findings from 1990 to 2006 from the same domain-concept. Please note that n/a values in columns (*Ever been told blood pressure high*: *yes*) and (*Ever been told blood cholesterol high*: *yes*) are due to one of the following reasons: the support value of a finding is lower than the threshold value of 10% or a finding was not required to be asked in a particular year.

Table 6.8. Selected Findings from BRFSS 1900 to 2006

Years	**(*Ever been told blood pressure high*: *yes*)***	**(*Ever been told blood cholesterol high*: *yes*)**	**(*BMI*: *overweight*) ****	**(*BMI*: *obese*)**
1990	45.9	n/a	52.8	n/a
1991	48.4	29.6	51.4	n/a
1992	51	31.9	51	n/a
1993	47.5	37.1	47.4	n/a
1994	n/a	n/a	46.5	n/a
1995	42.5	36.2	43.8	n/a
1996	15	n/a	51.2	n/a
1997	56.6	39.4	52.3	n/a
1998	11.8	n/a	55.1	n/a
1999	58.5	41.7	57.1	n/a
2000	n/a	n/a	33.7	41.6
2001	64.2	47.4	33.5	42.9
2002	11.2	10.1	32.7	43.8
2003	65.3	51.9	31.6	46
2004	12.3	n/a	31.7	46.8
2005	67.3	57.4	31	47.9
2006	n/a	n/a	34.71	24.75

*The fluctuation of the values is caused by whether the question is asked for all states in particular years.

**From 1990 to 1999, (*BMI*: *Overweight*) was defined to be (BMI $\geq$ 27.8) for males and (BMI $\geq$ 27.3) for females. Hence, there was no (*BMI*: *Obese*) reported during that time. Since BRFSS 2000, (*BMI*: *Overweight*) and (*BMI*: *Obese*) have been (25.0 $\leq$ BMI < 30.0) and (30.0 $\leq$ BMI < 99.99), respectively.

6.3.4 DCMiner Implementation

The BRFSS 2000-2006 DCM mining results of the domain-concept (*diabetes*: *yes*) with both tabular (text) and graphical formats are offered by DCMiner. The main page of the search system is shown in Figure 6.4.

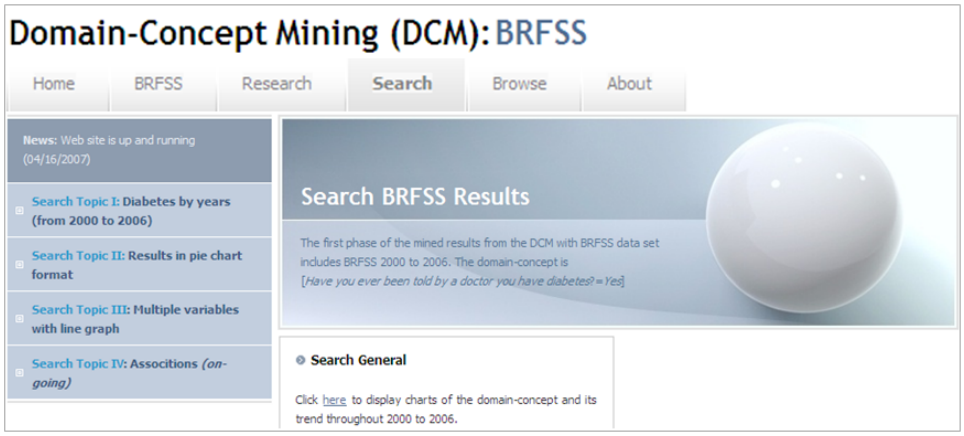


Figure 6.4. DCMiner for BRFSS 2000-2006.

Figure 6.5 is an example of a graphical format, which shows a trend of the distributions (percentage) of the domain-concept (*diabetes*: *yes*) in the BRFSS 2000 – 2006. DCMiner also allows the experts to select multiple (*attribute*: *value*) pairs. In this example, the selection is (*general health status*: *fair*) AND (*healthcare coverage*: *no*) to plot their percentages for trends as shown in Figure 6.6.

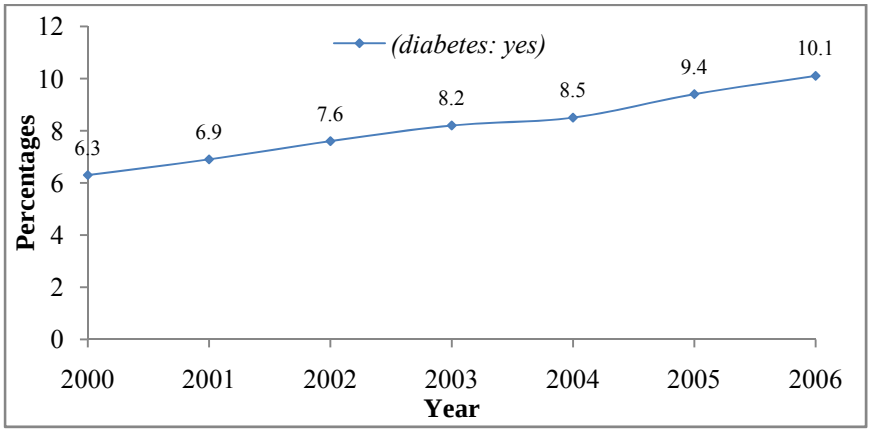


Figure 6.5. The distributions of (*diabetes*: *yes*) in BRFSS 2000 - 2006.

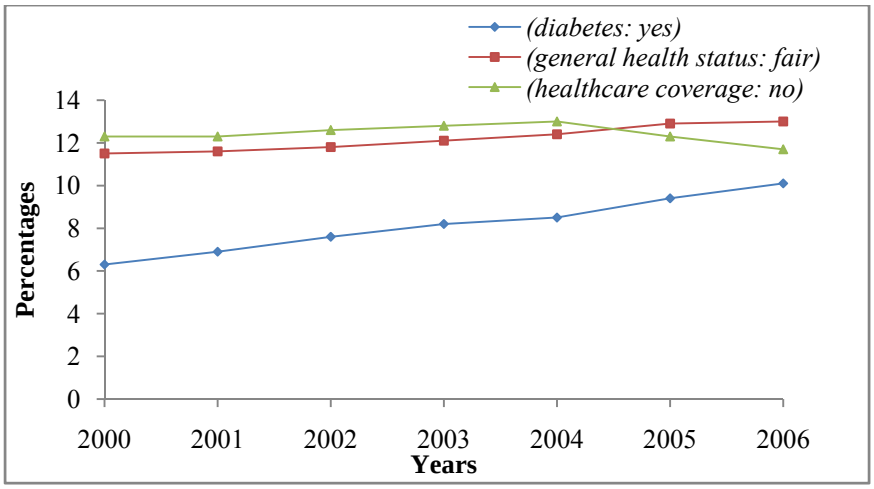


Figure 6.6. Comparisons among trends of percentages of each (*attribute*: *value*) offered by DCMiner.

The results of another search functionality of DCMiner are shown in Table 6.9, which contains a set of results with multiple selections: (*General Health Status*: *fair*) AND (*Healthcare Coverage*: *no*). The same results are represented as a histogram (Figure 6.7), which is also offered by DCMiner.

Table 6.9. An Example of a DCM Tabular Format Representation of the Results when Search for (*General Health Status*: *fair*) AND (*Healthcare Coverage* : *no*) within the Domain-Concept (*diabetes*: *yes*)

Domain-Concept	**Number of Transactions**	**Probability (of 355,710 transactions total in BRFSS 2006)**	**Conditional Probability {given (*diabetes*: *yes*)}**
(*diabetes*: *yes*)	36,985	0.1	n/a
(*General Health Status*: *fair*) AND (*Healthcare Coverage* : *no*)	41,492	0.12	n/a
(*General Health Status*: *fair*) AND (*Healthcare Coverage* : *no*) GIVEN (*diabetes*: *yes*)	1,037	0.003	0.3

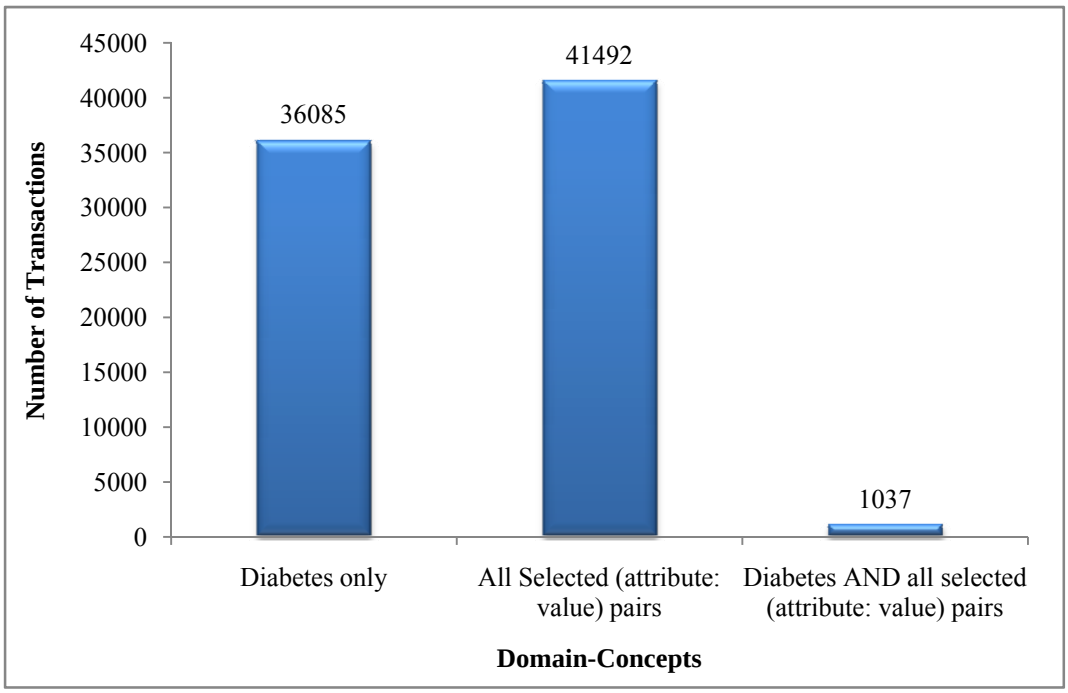


Figure 6.7. Histogram representation offered by DCMiner of the search results shown in Table 6.9.

The BRFSS 2006 DCM mining results of various domain-concepts (beyond diabetes) with tabular (text) formats offered by DCMiner with DCM-PA functionalities can be accessed on-line. The main page of the system is shown in Figure 6.8 (a). The same figure also contains an example of findings (itemsets) of size 1 to 4 (as shown in Figures 6.8 (a) – (e), accordingly) when a human expert select (*diabetes*: *yes*) as the domain-concept. Please note that we chose to display this domain-concept for continuity.

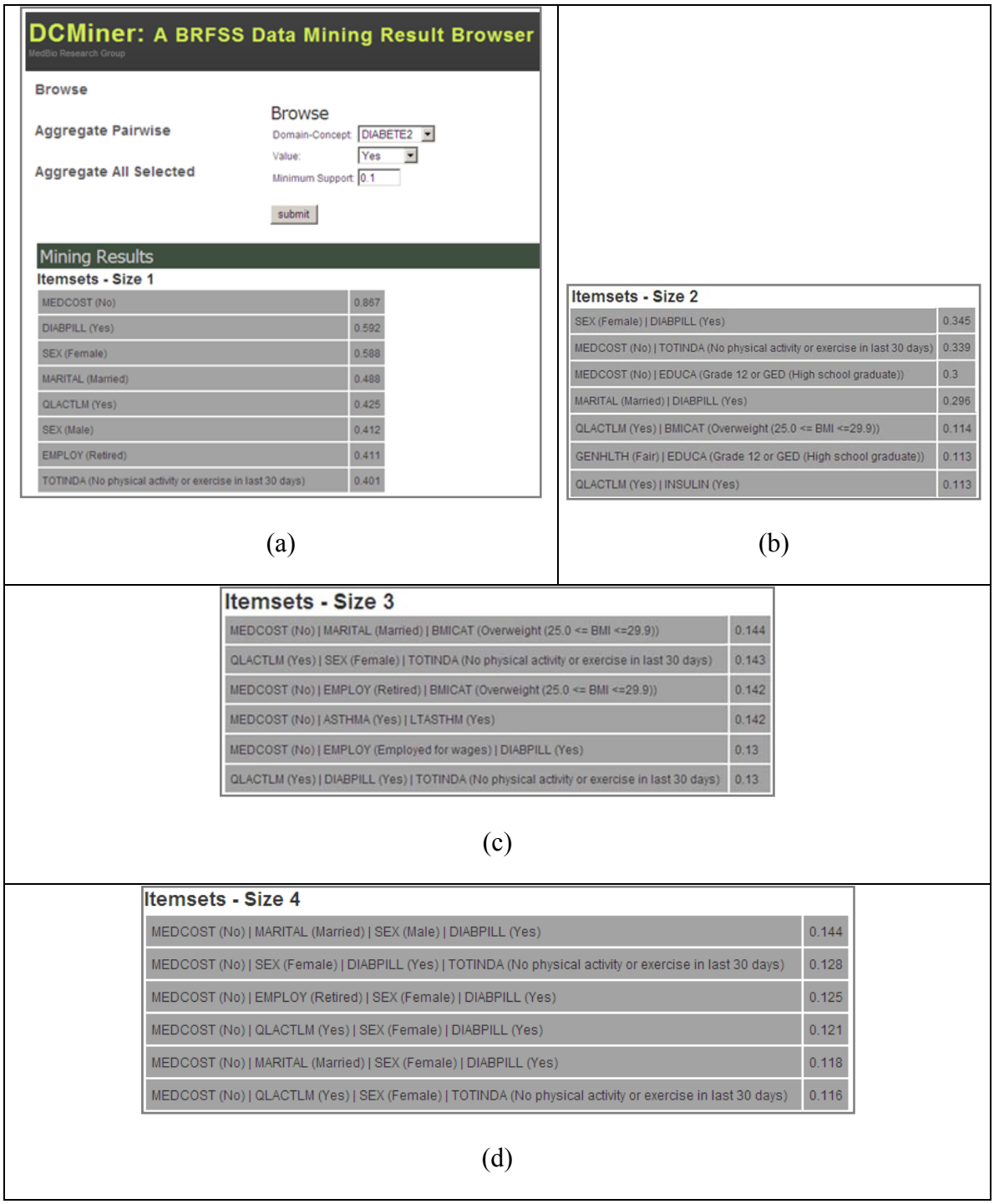


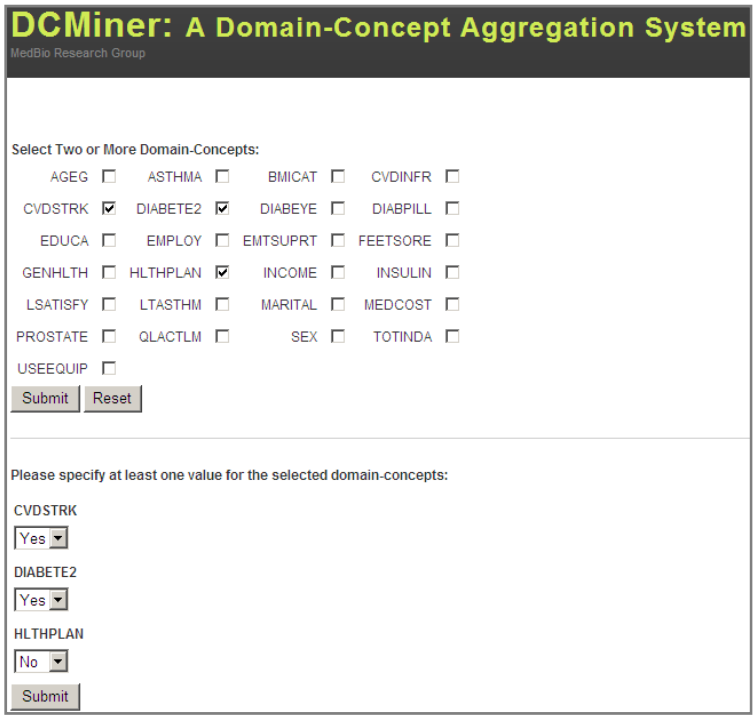
Figure 6.8. BRFSS's DCMiner result browser (tabular format) for dc partition (*diabetes*: *yes*) with associations (itemsets) of sizes (a) 1, (b) 2, (c) 3, and (d) 4.

6.3.4.1 BRFSS with Domain-Concept Partition Aggregation

Beyond the work presented above, BRFSS 2006, in particular, is utilized to demonstrate the DCM-PA approach, which is detailed in Chapter 4. Also, the reports of the experimental results and evaluation of DCM and DCM-PA are presented in Chapter 5. It is important to note that the purpose of DCM-PA is to give human experts the flexibility to aggregate (union and/or intersections) the findings from any number of domain-concepts. Figure 6.9 (a) illustrates the DCMiner with DCM-PA, which is located at http://medbio-ext.rnet.missouri.edu/brfss/aggregate. Further, an example of the aggregation results when the expert selects (*CVDSTRK* or *Have you ever been told by a doctor you have a stroke?*: *yes*), (*DIABETE2* or diabetes: *yes*), and (*HLTHPLAN* or *Do you have any healthcare coverage?*: *no*) is shown in Figure 6.9 (b). The details of the intersection and union probabilities and their calculations can be found in Chapter 4.

6.3.5 Conclusion and Future Work

Work on DCM and DCM-PA is an on-going endeavor in order to further include the other BRFSS data sets, and to provide the experts with a full set of visualization techniques (e.g. graphs and charts) through DCMiner. Further, we plan to conduct a systematic evaluation of the approach by health informatics experts, biostatisticians and other professionals with relevant expertise. The objective of the evaluation will be to determine: 1.) the exploration of relationships among BRFSS variables, 2.) the quality evaluation of findings, and 3.) the usefulness of DCMiner.



(a)

An aggregation among CVDSTRK (Yes) DIABETE2 (Yes) HLTHPLAN (No)	Intersection	Union
GENHLTH (Fair)	0.304	0.232
MEDCOST (No)	0.465	0.736
ASTHMA (Yes)	0.240	0.152
QLACTLM (Yes)	0.599	0.336
MARITAL (Married)	0.373	0.442
MARITAL (Divorced)	0.309	0.174
EDUCA (Grades 9 through 11 (Some high school))	0.194	0.118
EDUCA (Grade 12 or GED (High school graduate))	0.392	0.365
EDUCA (College 1 to 3 years (Some college or technical school))	0.194	0.253
EDUCA (College 4 years or more (College graduate))	0.101	0.185
EMPLOY (Employed for wages)	0.171	0.337
INCOME ($15,000 to $19,999)	0.171	0.119

(b)

Figure 6.9. DCM-PA (a) interface (a menu to aggregate all selected domain-concepts), and (b) aggregation results.

Finally, we plan to offer a complete system through which human experts can submit new (incremental) data sets online to be efficiently mined by DCM. Subsequently,

DCM-PA adaptively aggregates the new with the pre-existing mining results. And then, the experts may instantly browse the results through DCMiner.

6.4 Breast Cancer Survivors with Lymphedema

Preliminary research using database management and statistical analysis has been conducted to a data set of 202 participants, who are breast cancer survivors in the National Institute of Health (NIH)-funded 30-month post-breast cancer diagnosis study. The research will serve as a ground work prior to an implementation of the DCM approach, which aims to uncover associations among the following categories: 1.) limb volume change, 2.) BMI change, 3.) sign and symptom (such as pain, weakness, tenderness, redness, blistering, among others), 4.) medication history (includes prescription and non-prescription drugs), and 5.) breast cancer treatment (such as radiotherapy, chemotherapy, surgery, etc.).

The unique contribution of this preliminary work is a proposed *5% BMI-adjusted limb volume change* (LVC) approach, which considers a change of 5% or greater in breast cancer affected-arm volume over percent change in BMI, to be indicative of lymphedema. The research aims to: 1.) identify lymphedema development risks that may be associated with BMI-adjusted LVC, dominant limb and cancer-affected side and post-op swelling (1~4 weeks after surgery), and 2.) show an importance of pre-op (before surgery) limb measurement as a reference for detection of limb swelling. The research details and findings are as follows.

6.4.1 Post-Op Swelling and Lymphedema Following Breast Cancer Treatment: A Baseline-Comparison BMI-Adjusted Approach

Over 200,000 American women and over one million women around the world are newly affected by breast cancer each year [99, 100]. The two million breast cancer survivors living in the US and ten million worldwide are at lifetime risk for the development of lymphedema [101, 102], a chronic condition involving accumulation of protein-rich fluid which impacts physical, functional, and psychosocial health and well-being [85-89]. Second only to breast cancer recurrence, lymphedema is the most dreaded sequalae of breast cancer treatment [103].

The percentage of breast cancer survivors who develop lymphedema is not precisely known, although it is conservatively estimated that as many as 50% of survivors may experience lymphedema during their lifetimes [87-89]. The discrepancy between the reported percentages of 3% to 62.5% [88-90] in the literature stems from difficulties in measurement, diagnosis, and follow-up [104-109]. Common quantitative criteria for lymphedema include: two or more centimeters difference in limb girth between the affected and non-affected limbs; a 200 ml limb volume difference; or a 10 percent limb volume change (LVC) [104, 110].

The reported incidence fluctuates greatly among groups of individuals at risk for lymphedema [111, 112]. Although a number of factors have been implicated as associated with increased risk of lymphedema, including axillary dissection, radiation therapy, post-op infection, age, and weight gain [106, 109, 113-117], the diagnostic criteria themselves require further refinement in order to clarify actual occurrence of

lymphedema [104]. One of the dilemmas of the current afore-mentioned anthropometric criteria for lymphedema is that they are not calibrated to account for selected individual changes which commonly occur over the course of breast cancer treatment, such as fluid retention and changes in body mass index (BMI) [118].

Just as it is identified that increased BMI is associated with higher risk of breast cancer and poorer outcomes [119], including breast cancer recurrence [120], second primary cancers, and higher morbidity and mortality [121, 122], studies have identified a correlation between both BMI and BMI change and the development of lymphedema after breast cancer treatment [123, 124]. Unfortunately, the 2 cm, 200 ml, and even 10% LVC criteria do not take into account the changes experienced in the body that result in weight gain during or following treatment. The aim of this study was to develop and refine a BMI-adjusted criterion for lymphedema occurrence [118] that would take into account the commonly-experienced fluctuations in weight during and following breast cancer treatment.

6.4.1.1 Methods

In this National Institute of Health (NIH)-funded prospective repeated-measures study, 202 breast cancer survivors were recruited to participate in the 30-month study starting from pre-op visit (visit T0 after breast cancer diagnosis and before surgery). Participants were seen at post-op examinations, every 3 months for 12 months, and then every 6 months for 18 months for a total of 30 months (see Figure 6.10). Of all participants, 193 were unilateral breast cancer survivors. From this group, there were 105 participants whose cancer-affected side was their dominant limb (11 participants were

left-handed, 94 participants were right-handed); whereas, there were 88 participants whose cancer-affected side was not their dominant limb. From the same group of 193 participants, there were 37 participants who experienced swelling during the post-op visit (visit T1).

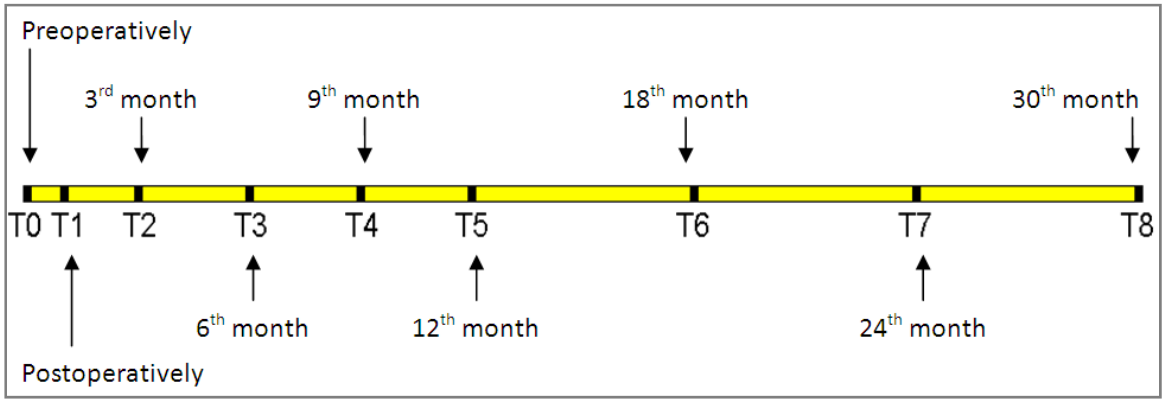


Figure 6.10. Timeline for data collection (pre-op to 30 months following surgery).

Arm circumferences were measured at every 4 cm using non-stretch tape measures [105, 125]. Volume was calculated using a summation of cylinder volumes (v), a derived cylinder formula shown in equation (6.4). Please note that a cylinder's base area was inferred from an average of two circular areas associated with two consecutive circumference measurements (c_1 and c_2) starting from the wrist to the underarm; and a cylinder's height is 4 cm.

$$v = \frac{1}{2\pi}(c_1^2 + c_2^2) \tag{6.4}$$

The unique contribution of this research is a proposed 5% BMI-adjusted limb volume change (LVC) criterion which is a potentially valid and reliable measure of lymphedema. Because: 1) increased BMI is associated with higher risk of lymphedema occurrence following breast cancer; 2) current standards rarely consider simultaneous

contralateral LVC; and 3) study participants' BMI ranged from 17.2 to 54.4 (average 30.5), BMI was considered in the assessment for lymphedema occurrence. BMI was calculated using the formula and categories of the Centers for Disease Control and Prevention [126] (Table 6.10). The same table also shows the percentages of women in this study per BMI category compared to women age 18 years or older who answered the *Behavioral Risk Factor Surveillance System* (BRFSS) survey in year 2006 [33]. The BRFSS 2006 data showed that women in the state of Missouri (MO) had higher percentages for the overweight and obese categories than the national percentages. These statistics are consistent with the higher percentages of the same BMI categories among the participants in this study.

Table 6.10. Adult Women BMI Weight Status

BMI	Weight Status	Percent of Participants	BRFSS 2006	
			Percent of MO Women	Percent of US Women
Below 18.5	Underweight	1.6%	37.3%	40.3%
18.5 – 24.9	Normal	22.8%		
25.0 – 29.9	Overweight	31.6%	30.1%	28.7%
30.0 and Above	Obese	44%	28.9%	24%

6.4.1.2 Analysis

Occurrence of lymphedema was first calculated from percent change in cancer-affected limb volume at each of eight post-op time points (starting from 1~4 weeks to 30 months post-surgery) compared to pre-op (before surgery) limb volume. Secondly, percent change in BMI during the same time periods were calculated. Finally, a change of

5% or greater in affected-arm volume over percent change in BMI was considered to be indicative of lymphedema. Two sets of statistical analyses were conducted between: 1) the cancer-affected dominant and non-dominant limbs; and 2) those with and without post-op (1~4 weeks after surgery) swelling. Unpaired (two-sample or independent-samples) t-tests were used to determine statistical significance [72, 127]. Relative risk was calculated to estimate the magnitude of the difference.

Participants were grouped according to their BMI weight status [126], as shown in Table 6.10. To find whether there was an increased risk of developing lymphedema on the dominant limb side that may be used more often, the first analysis compared risks of developing lymphedema from 3 months to 30 months post-surgery (visits T2 to T8) between the group of participants whose cancer affected their dominant limb side and the group of participants whose cancer affected their non-dominant side.

To find whether there was an increased risk of developing lymphedema that may be associated with the swelling caused by breast cancer surgery, the second analysis compared risks of developing lymphedema during the same time period as the first analysis (visits T2 to T8) between the group of participants who met or exceeded the *5% BMI-adjusted LVC* criterion at the post-op visit (visit T1) and the group that did not meet this criterion at visit T1.

6.4.1.3 Results

For all unilateral cancer-affected limb participants (n = 193), 63% (n = 121) met the *5% BMI-adjusted LVC* criterion at some point following (excluding) the post-op visit (mean time to criterion = 9 months, standard deviation = 7 months).

6.4.1.3.1 Cancer-Affected Dominant and Non-Dominant Limbs

To answer the question of whether there was an increased risk of developing lymphedema when a patient's cancer-affected limb was her dominant side, t-test and relative risk analyses were used to compare between two groups of participants: 1) cancer-affected dominant limb group; and 2) cancer-affected non-dominant limb group. Overall, the relative risk between these groups was 1.1, and there was not a significant difference (65.7% compared to 59.1%; t = 0.95; p = 0.35) as detailed in Table 6.11 (see also Figure 6.11).

Table 6.11. Relative Lymphedema Risk Analysis between Cancer-Affected Dominant and Non-Dominant Sides

Dominance, Cancer-Affected Side, and Lymphedema		**BMI Status**				**Total**
		Underweight	**Normal weight**	**Overweight**	**Obese**	
Cancer-Affected *Dominant* Limb	Total Number of Participants	1	23	35	46	105
	Swelling at Visits T2 to T8	0 of 1 (0%)	12 of 23 (52.2%)	22 of 35 (62.9%)	35 of 46 (76.1%)	69 of 105 (65.7%)
Cancer-Affected *Non-Dominant* Limb	Total Number of Participants	2	21	26	39	88
	Swelling at Visits T2 to T8	1 of 2 (50%)	13 of 21 (61.9%)	18 of 26 (69.2%)	20 of 39 (51.3%)	52 of 88 (59.1%)

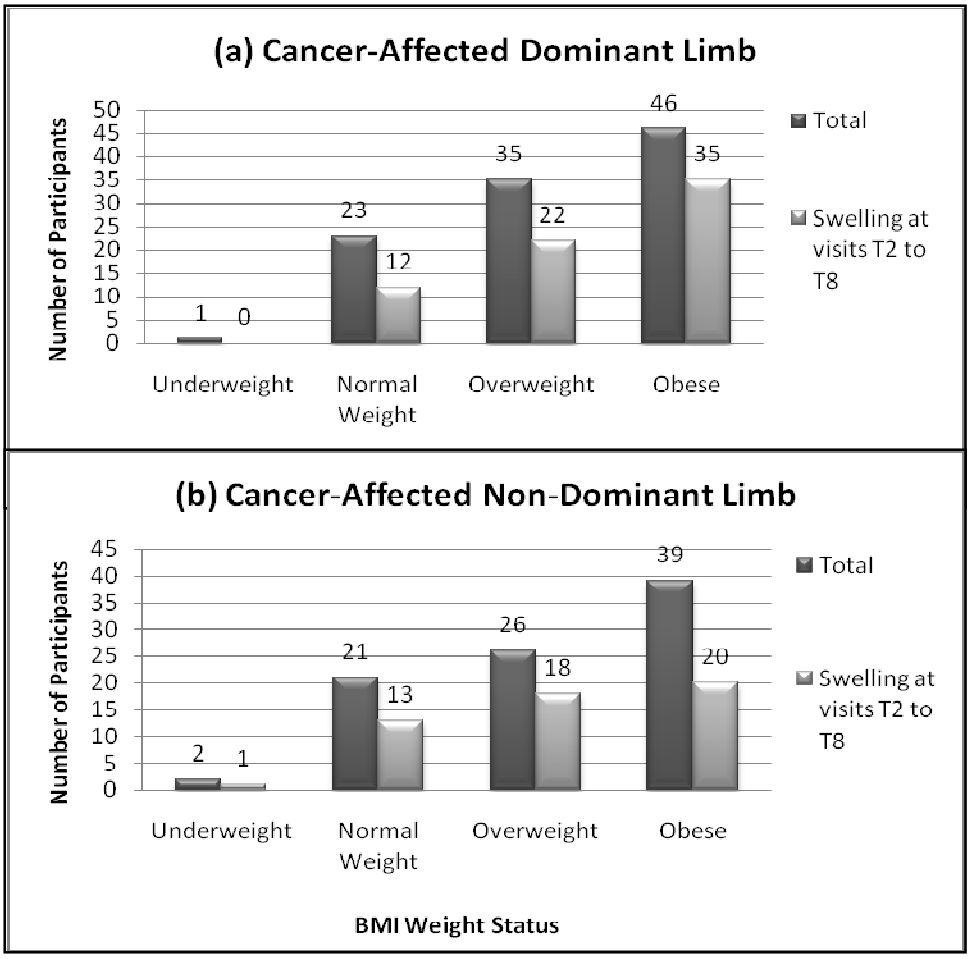


Figure 6.11. Number of participants whose: (a) cancer affected their dominant limb; or (b) cancer affected their non-dominant limb, categorized by BMI weight status.

Tests of statistical significance were also conducted for lymphedema occurrence and non-occurrence in the three BMI categories—normal weight, overweight, and obese, with the reported relative risks of 0.84, 0.91, and 1.48, reported t values of 0.64, 0.51, and 2.44, and reported p values of 0.53, 0.61, and 0.02, respectively. Even though in the larger group analysis, limb dominance and cancer-affected side were not significantly associated with the risk of developing lymphedema, participants with BMI 30 and above had significantly higher risk of developing lymphedema if their cancer treatment was on

the dominant side (rr = 1.48, 48% higher risk). Please note that the underweight group was not tested due to its small sample size.

6.4.1.3.2 With and Without Post-op Swelling

A relative risk analysis was calculated to compare the risk of developing lymphedema at later visit (visits T2 to T8) between the groups of participants with and without post-op (visit T1) swelling. Overall, the relative risk between these two groups was 1.4, and there is a significant difference between the groups (81.1% compare to 58.3%; t= 2.6; p = 0.01) as detailed in Table 6.12 (see also Figure 6.12). Those with post-op swelling had a 1.4 greater risk of developing lymphedema at some later point, as compared to those without post-op swelling.

Table 6.12. Relative Lymphedema Risk Analysis between Participants with and without Post-op Swelling

With and Without Post-Op Swelling		BMI Status				Total
		Underweight	Normal Weight	Overweight	Obese	
Total Number of Participants		3	44	61	85	193
Post-Op Swelling	Swelling at Post-Op Visit	0 of 3 (0%)	9 of 44 (20.5%)	13 of 61 (21.3%)	15 of 85 (17.6%)	37
	Swelling at Visits T2 to T8	0 of 0 (N/A %)	5 of 9 (55.6%)	11 of 13 (84.6%)	14 of 15 (93.3%)	30 of 37 (81.1%)
***No* Post-Op Swelling**	No Swelling at Post-Op Visit	3 of 3 (100%)	35 of 44 (79.5%)	48 of 61 (78.7%)	70 of 85 (82.4%)	156
	Swelling at Visits T2 to T8	1 of 3 (33.3%)	20 of 35 (57.1%)	29 of 48 (60.4%)	41 of 70 (58.6%)	91 of 156 (58.3%)

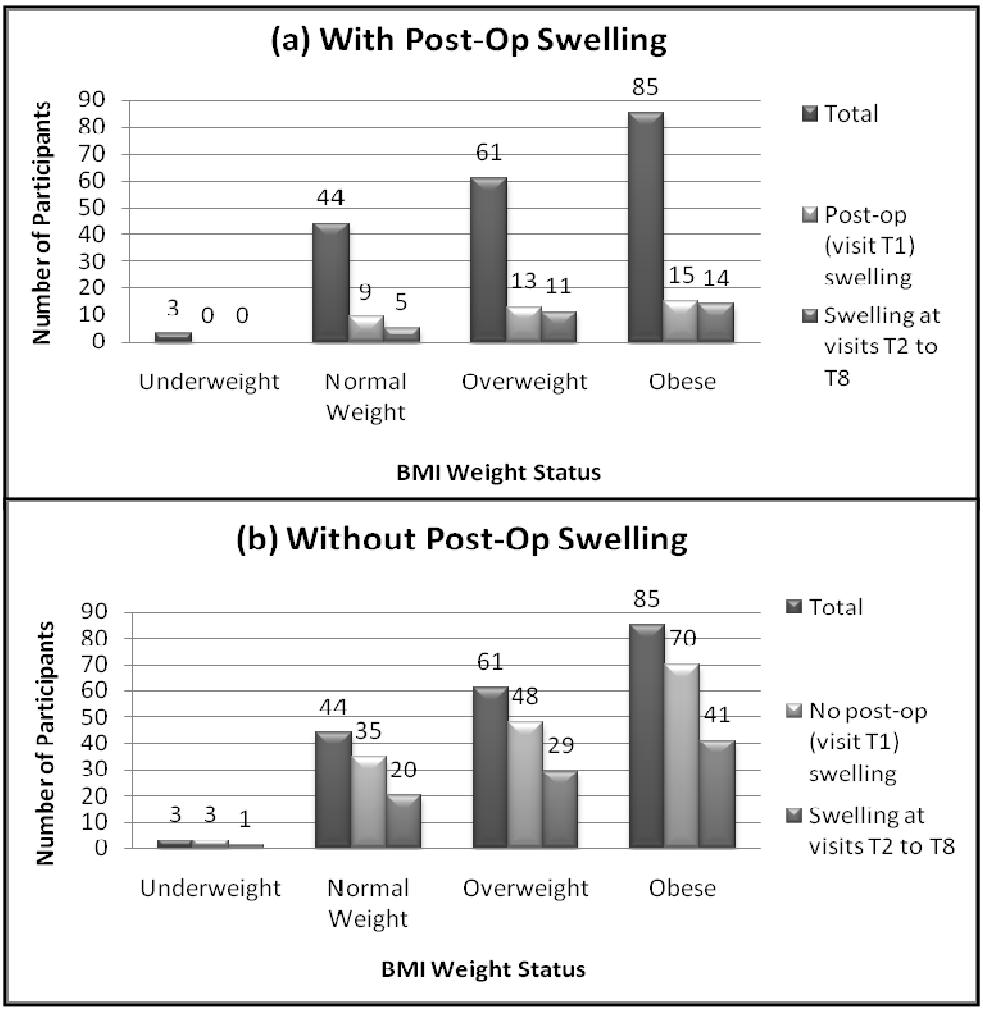


Figure 6.12. Categorized by their BMI status, number of participants who: (a) experienced post-op swelling, or (b) did not experience post-op swelling**.**

Tests of statistical significance were also conducted for each of the three BMI categories—normal weight, overweight, and obese, with the reported relative risk values of 0.97, 1.4, and 1.6, reported t-values of 0.08, 1.64, and 2.63, and reported p values of 0.93, 0.11, and 0.01, respectively. In addition to the larger group analysis in which post-op swelling was significantly associated with the risk of developing lymphedema, a sub-category analysis revealed participants with BMI above 25 had a higher relative risk of developing lymphedema than the normal weight group (rr = 1.4 and 1.6, 40% and 60%

higher risk for overweight and obese, respectively). Please note that the underweight group was not tested due to its small sample size.

Further analyses showed the significance of having the pre-op (before surgery) measurement when we substituted the data from the 3-month (following surgery) visit for the missing pre-op data. Without the pre-op measurement, 49 participants who met the *5% BMI-adjusted LVC* criterion at visit T2 would not have been recognized.

6.4.1.4 Discussion

For all participants, 63% met the *5% BMI-adjusted LVC* criterion at some point following (excluding) the post-op visit. Limb dominance and cancer-affected side were not significantly associated with the development of post-surgery lymphedema (relative risk = 1.1) in the group as a whole. In the subgroup analysis, those with higher BMI showed a 48% greater lymphedema risk in women whose cancer occurred on their dominant side. Further, post-op swelling significantly increased the risk of later developing lymphedema (relative risk 1.4) across the group as a whole. This means the person who developed post-op swelling was 40% more likely to develop lymphedema at some later time (before 30 months) after surgery. In the subgroup analysis, this relative risk of developing lymphedema was even higher in the overweight and obese BMI groups than for normal weight women (40% and 60% greater risk).

Also of importance, among 121 participants who later met the *5% BMI-adjusted LVC* criterion, there were 49 participants with lymphedema who would have been overlooked if the pre-op measurements were not available. Further, since post-op swelling is associated with higher risk of developing lymphedema, having the pre-op

baseline is an essential reference for detection of post-op swelling. This finding documents the need for pre-op assessment in the clinical setting.

6.4.1.5 Conclusions

Using the *5% BMI-adjusted LVC* approach to assessment of lymphedema occurrence provides the opportunity for a more valid and reliable estimation of post-breast cancer lymphedema occurrence. Also important is the capability to compare pre-op limb volume measurements to post-op volume. Based on this preliminary analysis, lymphedema is a risk for approximately two-thirds of breast cancer survivors in the 30 months after surgery. These data suggest increased risk for lymphedema in survivors with higher BMI whose dominant limb was treated for cancer. Overall, breast cancer survivors with post-op swelling have a significantly higher risk of developing lymphedema than those who do not have post-op swelling. It is the group with higher BMI who have the greatest risk of developing lymphedema. Breast cancer survivors with higher BMI appear to have cumulative risk of developing lymphedema if the cancer was on the dominant side or if they experience post-op swelling. The survivors who are overweight or obese will benefit from education on maintaining optimal BMI and lymphedema risk reduction practices, as well as careful monitoring for limb and symptom changes. Further vigilance is required for participants with higher BMI who have cancer treatment to the dominant side or experience post-op swelling.

Further research to examine the constellation of risk factors that contribute to the development of lymphedema in breast cancer survivors must include the consideration of:

1.) Pre-diagnosis BMI,

2.) BMI increase in survivorship,

3.) Occurrence of post-op swelling, and

4.) Cancer treatment to the dominant side.

Increased understanding of the cumulative impact of these and other known risk factors will enable researchers and clinicians to design and implement more targeted risk-reduction interventions.

6.4.2 DCM and Its Web-Based System for Lymphedema

The DCM Web-based interfaces for lymphedema research are presented in this section. Figure 6.13 illustrates the main page of system. This page contains links to the other components of the system, which includes the search functionalities. Examples of the outputs from the search functionalities are shown in:

1.) Table 6.13 contains summaries of limb swelling sides and volume (cc) levels. The summaries are drawn from a search functionality, which lists the numbers of patients categorized by the volumes of limb swelling (starting from 50 cc to 200 cc) and the cancer-affected sides (left, right, or both arms). Please note that for those participants whose cancer-affected sides are both arms (B), the limb swelling side can be left only (L), right only (R), or both sides.

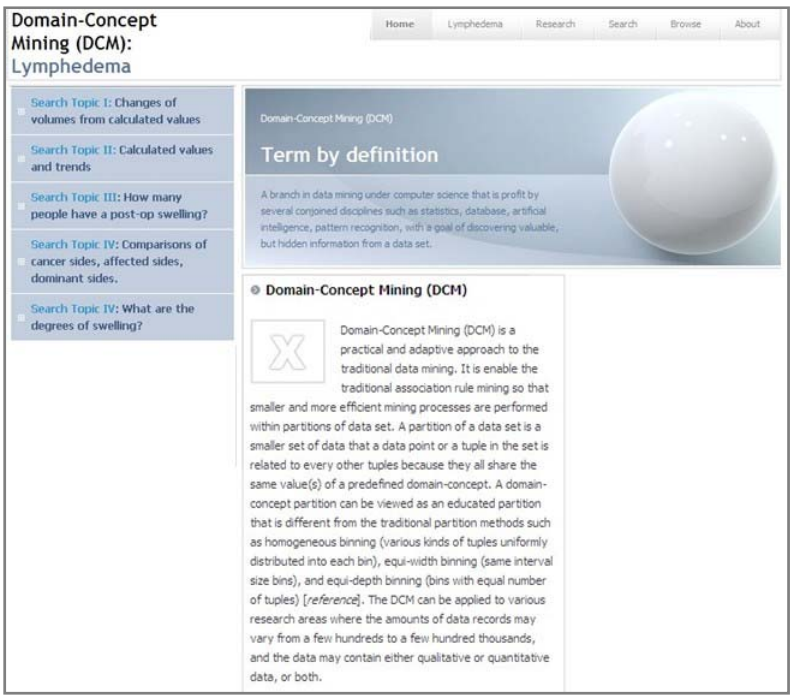


Figure 6.13. Main page of the DCM web system for the lymphedema data set.

Table 6.13. Summaries of Limb Swelling Sides and Volume (cc) Levels

Cancer-Affected Side	Volume (cc) Level	Number of Participants
Left	50	43
	100	36
	150	27
	200	45
Right	50	47
	100	48
	150	37
	200	49
Both (Limb Swelling on Both)	200	2
Both (Limb Swelling on Left)	50	1
	100	1
	200	2
Both (Limb Swelling on Right)	100	1
	200	2

2.) Figure 6.14 is an alternative representation of the information shown in Table 6.13.

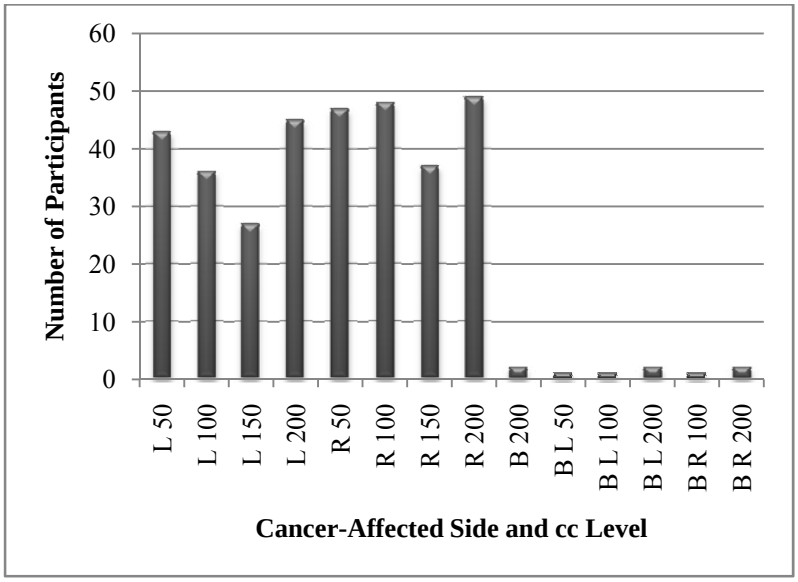


Figure 6.14. Summaries of cancer-affected sides and limb swelling (cc) levels.

3.) Figure 6.15 offers the human expert a comparison between the numbers of participants categorized by cancer-affect sides, and cancer-affected *dominant* side.

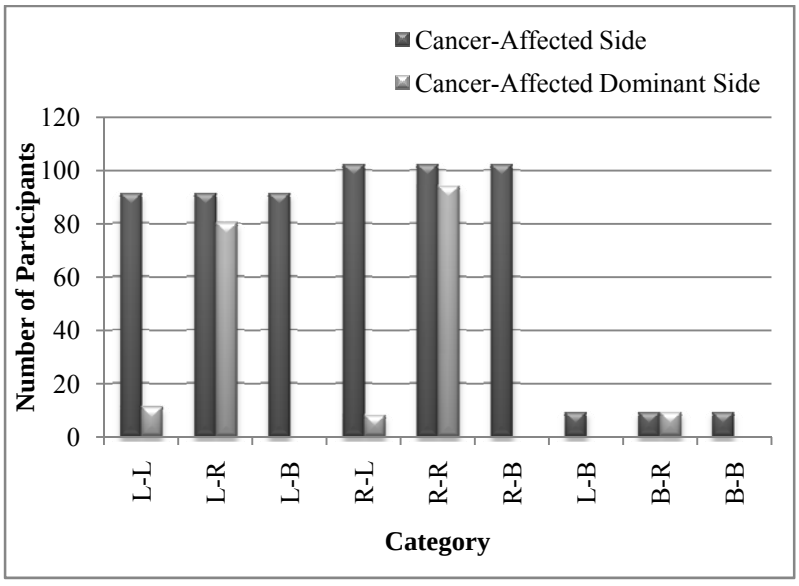


Figure 6.15. A comparison between: 1.) cancer-affect sides and 2.) cancer-affected dominant sides.

6.5 Domain-Concept Mining for Large-Scale and Complex Cellular Manufacturing Tasks

To provide a novel domain-concept mining (DCM) algorithm that offers solutions to complex cell formation problems, which consist of a non-binary machine-component (m/c) matrix and production factors for fast and accurate decision support. The DCM algorithm first identifies the domain-concept from the demand history and then performs association rule mining to find associations among machines. After that, the algorithm forms machine-cells with a series of inclusion and exclusion processes to minimize inter-cell material movement and intra-cell void element costs as well as to maximize the grouping efficacy with the constraints of *Bill of Material* (BOM) and the maximum number of machines allowed for each cell.

The DCM algorithm delivers either comparable or better results than the existing approaches using the known binary data sets. We demonstrate that the DCM can obtain satisfying machine-cells with production costs when extra parameters are needed. The DCM algorithm adapts the idea of the *Sequential Forward Floating Selection* (SFFS) [128] to iteratively evaluate and arrange machine-cells until the result is stabilized. Even though, the SFFS algorithm is an improvement over a greedy algorithm called *Sequential Forward Selection* (SFS) [129, 130], SFSS can only ensure sub-optimal solutions. However, the machine-cells problem is considered NP-hard [73], and thus, achieving sub-optimal solutions within a certain computational complexity limitation may be acceptable. The DCM algorithm considers a wide range of production parameters, which make the algorithm suitable to the real-world manufacturing system settings.

The proposed DCM algorithm is unlike other array-based algorithms. It can group non-binary m/c matrix with considerations of real-world factors including product demand, BOM, costs, and maximum number of machines allowed for each cell.

6.5.1 Motivations

Due to changes of customers' demand pattern in contemporary market places, traditional fixed production lines that produce very large batches of products with long production lead-time have been becoming out-of-date shop floor plans. Modern manufacturing entities must adopt flexible production approach to accommodate the challenge from competitive and changing market. The Flexible Manufacturing System (FMS) has been emerging as an essential concept to conform the task to cluster flexible facility assemblages for small batches that can rapidly respond to changes for different product orders and design changes.

Cellular Manufacturing (CM) is an effective approach for determining functional machine layouts when sequential production lines are no longer practical in small-median batch manufacturing environments. CM posits a common management principle: grouping related manufacturing tasks such that tasks with similar requirements are associated within the same work cells [131]. In CM, the manufacturing facilities are divided into "cells" where distinctive functional machines produce a family of products or parts. Grouping machines and parts according to the ideas of Group Technology (GT) is a natural starting point of CM and cell formation - the fundamental problem of CM system design. Given the entire set of parts and available machines, the objective of cell formation is to configure a set of machine-cells and a partition of parts which streamlines

the production flow. By devoting a machine-cell to the manufacturing of a part family, advantages have been reported in many aspects, such as setup time reduction, work-in-process reduction, throughput time reduction, material handling costs reduction, scheduling simplification, and product quality improvement [132].

It is well recognized that simply grouping the machines from a binary machine-component (m/c) incidence matrix is a far cry from real-world situations; other important manufacturing factors should also be considered and recorded in the matrices. Additionally, to mimic the real-world setting, hierarchies of components should be included in the decision making process. Figure 6.16 depicts an example of such hierarchies, and Table 6.14 shows the corresponding Bill of Material (BOM) matrix. The numbers in the BOM matrix represent the amount of components needed for parent components in the hierarchical structure. Figure 6.16 (a), (b), and (c) show the hierarchies of components to produce final products, *Pa*, *Pb*, and *Pc*, respectively. For example, to produce one unit of a final product *Pa* in Figure 6.16 (a), three *P1* and four *P2* are needed. This also suggests the operation sequence of *Pa*.

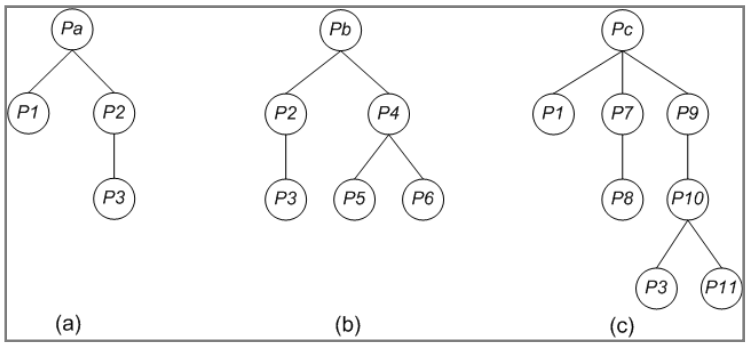


Figure 6.16. An example of a Bill of Material for three final products – *Pa, Pb*, and *Pc*, where (a), (b) and (c) show components needed and production sequences to produce *Pa*, *Pb*, and *Pc*, respectively.

In general, a component may be needed to produce various components and/or final products, such as three units of *P1* are needed to produce a unit of *Pa*, and nine units of *P1* are needed to produce a unit *Pb*. The second factor to be considered is the production cost matrix, which includes an aggregation of laboring, material, and handling costs. One more factor to be considered is the maximum number of machines allowed for each machine-cell, which can estimate the area required to locate the cell that a manufacturing facility will have to handle. The abovementioned matrices and factors should be utilized to generate an efficient production plan, possibly with alternatives for unexpected changes under some circumstances. The production plans will be used in the decision-making process that responds to the users' predefined criteria, which include the demands for products, possible machine breakdowns, and changes in production costs.

Table 6.14. A Bill of Material Matrix for Figure 6.16, where the Matrix Suggests Units of Components Needed to Produce Other Components and/or Final Products

Parts	***P1***	***P2***	***P3***	***P4***	***P5***	***P6***	***P7***	***P8***	***P9***	***P10***	***P11***
Final product *Pa*	3	4									
Final product *Pb*		5		2							
Final product *Pc*	9						1		1		
P1	0										
P2		0	5								
P3			0								
P4				0	2	5					
P5					0						
P6						0					
P7							0	7			
P8								0			
P9									0	8	
P10			2							0	4
P11											0

6.5.2 Background

Approaches that customarily deal with the cell formation problems can be briefly categorized into various methods such as: mathematical programming, array-based algorithms, hierarchical clustering algorithms, non-hierarchical clustering algorithms, and heuristics. Many mathematical models were recently proposed to deal with particular versions of cell formation problem [41-43]. The advantage of the mathematical programming is that the formulations are capable of considering a variety of manufacturing information, such as space limitation, alternative production sequence, and/or product demand. Harhalakis et al. [133], and Cao and Chen [134] represented the physical limitation of maximum number of machines per cell by a constraint or an upper bound. Balakrishnan and Cheng [135] proposed a two-stage method that took into consideration of rearrangement cost and product demand of multi-period planning horizons. Sofianopoulou [136] presented an implementation of cellular manufacturing, which was able to evaluate the alternative production scenarios by data envelopment analysis (DEA). However, the balance between modeling a meticulous manufacturing system and simplifying the computational complexity is always difficult to maintain. Thus, finding comprehensive and yet feasible approaches is still a challenging research problem.

Compared to mathematical programming approaches, array-based algorithms, which solve the binary m/c matrix problem, are relatively efficient in terms of computational complexity and feasibility. The machine-cells and part families can be obtained simultaneously on the main diagonal of the m/c matrix by rearranging the matrix, where the columns are in accordance with parts and the rows are in accordance

with machines. Unlike the complex mathematical approach, the m/c matrix only provides limited binary information, (i.e. zero or one for each element), so important manufacturing information, such as product demands and inter-cell material movement costs, are rarely taken into consideration in the former algorithms, which could solve only the binary m/c matrix problems. There are three approaches, which utilize the binary m/c matrix, namely array-based, hierarchical clustering, and non-hierarchical approaches.

Array-based methods rearrange the rows and the columns in an m/c matrix in order to group the machines and the parts. An early contribution for array-based methods was made by Burbidge [137]. Array-based methods are a part of the Production Flow Analysis (PFA) procedure for the implementation of the cellular manufacturing system. A computational implementation of the PFA method, named GROUPTEC, and its case study has been reported by Santos and Araújo [138]. Other notable array-based methods include Rank Order Clustering (ROC) [139], Bond-Energy Algorithm (BEA) [140], and MODROC [141]. In contrast, hierarchical clustering methods use similarity or distance information to produce a hierarchy of clusters or partitions. Such methods are normally unable to arrange both machine-cells and part families simultaneously. Pioneering work on hierarchical clustering method was proposed by McAuley [142], and the most recent hierarchical clustering methods are GP-SLCA [132] and MOD-SLC [143]. Gupta [144] argued that the hierarchical clustering methods suffer from chaining effect problems. Similarly, non-hierarchical clustering approaches form machine-groups and part families by using similarity and distance functions. The number of clusters is often assigned a priori for non-hierarchical clustering algorithms. Chandrasekharan and Rajagopalan [145] proposed an ideal seed non-hierarchical clustering (ISNC) for binary cell formation

problems. Other common algorithms in this field include ZODIAC [146] and GRAFICS [147]. Miltenburg and Zhang [148] reported a comprehensive comparison and evaluation of many known algorithms including array-based, hierarchical clustering, and non-hierarchical clustering algorithms. Other approaches apply heuristics such as fuzzy logic [149], evolutionary algorithms [150], and genetic programming [151, 152] have been utilized to search a feasible solution. In addition to the abovementioned production-oriented algorithms, a hybrid manufacturing system (HMS) has been proposed to solve the cell formation problems by Zolfaghari and Roa [153]. The HMS is an integration of cellular manufacturing and job shop. The major advantage of the HMS approach is the ability of producing non-family part.

Recently, data mining techniques, such as association rules (AR) mining, have been applied to the same research problem. Chen [154] proposed an approach called *Association Rule Induction* (ARI) by applying the Apriori algorithm [12] from the data mining field to group machines. Although the abovementioned contributions are promising, the cell formation task is still challenging in actual practice because of large-scale machine-component relationships and difficulties in construction criterion functions. Therefore, a novel domain-concept association rules mining (DCM) algorithm is developed in this dissertation to solve large-scale and complex cell formation problems, where factors such as operation sequences, unit inter-cell material movement costs, demand for products, production quantities, and maximum number of machines allowed for each cell are considered for fast and accurate decision support.

A domain-concept is a partition mechanism for machines based on a prioritized factor list from the complex cell formation setting. For instance, if users would like to see

machines, which will be grouped within a cell, to produce components that are required for some products under some prioritized constraints (such as demands and a BOM), the DCM algorithm will mine rules from machines with parts that have high demands followed by machines with parts from a selected list in a BOM. Depending on the number of partitions in the domain, the DCM algorithm keeps mining secondary prioritized partitions and generates extra rules which will then be utilized for cell formation by their priorities

Rules mined from the DCM algorithm could be efficiently indexed in a database and utilized to meet the needs of decision support when unexpected changes happen. The details of the DCM algorithm, its architecture, and its criterion function are presented in the next section.

6.5.3 Methods

As shown in Table 6.15, the DCM algorithm first accumulates historical demand information (i.e. which product with what quantity) of each product into a demand vector (D), where D_i is a total demand value of P_i. The DCM also utilizes the predefined BOM matrix, as shown in Table 6.14, to calculate the total number of product needed to be produced, as shown at BOM row in Table 6.15.

Table 6.15. Demand Values (D_i), Resulting BOM Calculations by Applying D_i to the Values in Table 6.14, Predetermined Unit Inter-Cell Material Movement Costs (V_i), and Predetermined Unit Intra-Cell Void Element Costs (E_i)

Components	***Pa***	***Pb***	***Pc***	***P1***	***P2***	***P3***	***P4***	***P5***	***P6***	***P7***	***P8***	***P9***	***P10***	***P11***
D_i	18	35	15	3	2	2	1	4	8	5	2	6	4	5
BOM	18	35	15	(3*Pa*)+(9*Pc*)	(4*Pa*)+(5*Pb*)	(5*P2*)+(2*P10*)	2*Pb*	2*P4*	5*P4*	*Pc*	7*P7*	*Pc*	8*P9*	4*P10*
V_i	15	25	10	3	7	4	1	3	5	3	2	1	1	3
E_i	15	25	10	3	7	4	1	3	5	3	2	1	1	3

The algorithm uses these values instead of 0's and 1's in an m/c matrix. Each row of an m/c matrix is a machine, which is considered a transaction to be mined for association rules by the DCM. Each column is a component that is regarded as an attribute and may be identified as a domain-concept. Furthermore, the algorithm also accepts input matrices of the unit inter-cell material movement cost of each product (V_i) and the unit intra-cell void element cost (E_i) then minimizes these values when forming machine-cells.

It is essential to understand the basic idea of the AR mining that was introduced by Agrawal et al. [12] in order to better understand the DCM algorithm that served as the backbone of our decision making process. The pseudo codes of the DCM algorithm and its procedures are Figure 6.17. The AR mining statistically finds relationships among attributes of the underlying data without a prior knowledge or hypothesis. A discovered association rule $X \rightarrow Y$ tells that mutually disjoint sets of attributes X and Y co-occur with an observed frequency of X and Y happening at the same transactions. This frequency is called a support value. Moreover, a rule $X \rightarrow Y$ also carries a conditional probability of Y given X, which is called a confidence value. The confidence value indicates how often Y occurs when X occurs. To efficiently use the rules for a real-time decision support, we developed the DCM algorithm, which is an extension of the original AR algorithm.

DCM (parameters:C, M, D, MC, BOM, V, E, max_m)
1. Identify the x highest demand products, p_x, in D
2. (T_1, T_2, MC) = **BuildTransactions**(D, p_x,C, M, MC, BOM)
3. Execute AR mining to build rules of all machine pairs and obtain AR_{T_1} , AR_{T_2} , respectively.
4. FOR (each domain-concept dc, where dc = 1 to 2) {
5. WHILE (($MC \neq \phi$) AND ($AR_{T_{dc}} \neq \phi$)) DO
6. Form the tentative MG } }
7. Calculate $F(MG)$
8. (MG',$F(MG')$) = **AdjustCells**(MG, C, M, BOM, max_m)

9. WHILE $(F(MG') < F(MG))$ DO {
10. $MG = MG'$
11. $(MG', F(MG'))$ = **AdjustCells**(MG, C, M, BOM, max_m) }
12. RETURN (MG, $F(MG)$)

BuildTransactions (parameters: D, p_x, C, M, MC, BOM)

1. Set $count = 1, T_1 = \phi, T_2 = \phi, M' = M, C' = C, D' = D$
2. FOR (ALL $D_l \in D$) {
3. IF (p_x IN D_l) {
4. FOR (ALL p_j IN D_l) {
5. Filter BOM using p_j to obtain BOM_j
6. FOR (ALL C_i IN BOM_j) {
7. Identify the quantity needed for C_i. Add this number to *quant*.
8. Filter MC using C_i to obtain MC_i.
9. Create T_{count} by
10. Including C_i and
11. Including all machines in MC_i.
12. Update M' by excluding the previously selected M in MC_i
13. Update C' by excluding the previously selected C in MC_i
14. *count*++ }
15. FOR (q = 1 to *quant*) {
16. Add a new transaction T_{count} into T_1 }
17. Update the corresponding MC cells with *quant* }}
18. ELSE $D' = D' - D_l$ }
19. FOR (ALL $D_l \in D'$) {
20. Perform steps 4 to 17 for the machines in M' and the components in C' to build T_2. }
21. RETURN (T_1, T_2, MC)

AdjustCells(MG, C, M, BOM, max_m)

1. MG' = AdjustMachines(MG, M, max_m)
2. IF ($F(MG') < F(MG)$) {
3. MG' = AdjustComponents(MG' , C, BOM)
4. RETURN (MG' ,$F(MG')$) }
5. RETURN (MG, $F(MG)$)

AdjustMachines(MG, M, max_m)

1. FOR (ALL M_j in M){
2. Identify *original_cell* of M_j
3. Initialize *min_cost* to a large number
4. *selected_cell* = 0
5. FOR (ALL mg_k in MG) {
6. Calculate $F(M_j)$
7. IF ($F(M_j) < min\_cost$) {
8. $min\_cost = F(M_j)$
9. $selected\_cell = k$ }}
10. IF (($| mg_{selected\_cell} | < max\_m$) AND ($selected\_cell \neq original\_cell$)) {
11. Remove M_j from $mg_{original\_cell}$
12. Assign M_j to $mg_{selected\_cell}$

```
   13.    Update MG  }}
   14. RETURN (MG)


AdjustComponents(MG, C, BOM)
   1. FOR (ALL C_i in C) {
   2.    Identify original_cell of C_i
   3.    Initialize min_cost to a large number
   4.    selected_cell = 0
   5.    FOR (ALL mg_k in MG) {
   6.      Calculate F(C_i)
   7.      IF ((F(C_i) < min_cost) AND (C_i is from the same BOM as other C in mg_k) ) {
   8.        min_cost = F(C_i)
   9.        selected_cell = k }}
   10.   IF (selected_cell ≠ original_cell) {
   11.     Remove C_i from its mg_original_cell
   12.     Assign C_i to mg_selected_cell
   13.     Update MG  }}
   14. RETURN (MG)
```

Figure 6.17. Pseudo codes for the DCM algorithm, BuildTransactions, AdjustCells, AdjustMachines, and AdjustComponents procedures.

In general, the AR mining with domain-concept will report the associations among attributes within each domain-concept, with the support and confidence values, without considering the other criteria. The complex cell formation problem, on the other hand, has the knowledge for a certain set of machines that should be favorably grouped together to preserve the operation sequence and the hierarchical (a PART-OF) relationship among components shown in a BOM. The BOM suggests the DCM to make a decision to add only related components and their associated machines into a cell. Rules are then used to form efficient cells with regards to the total inter-cell material movement costs (V) and the total intra-cell void element costs (E). Both V and E directly determine the total cost (F) of the resulting machine-groups.

In this dissertation, we introduce a cost function F that the DCM attempts to minimize while forming machine-cells. The F function is expected to fulfill two objectives simultaneously: minimization of the inter-cell material movement cost and

maximization of the machine utilization when a new machine is added to a cell. F is defined as follows.

$$F(MG) = \sum_{i=1}^{|C|} \left(w_N V_i D_i N_i + w_G E_i G_i \right) \tag{6.5}$$

$$F(C_i) = \sum_{j=1}^{|M_{mg_k}|} (w_N V_{ij} D_{ij} N_j + w_G E_{ij} G_j) \tag{6.6}$$

$$F(M_j) = \sum_{i=1}^{|C_{mg_k}|} (w_N V_i D_i N_i + w_G E_i G_i) \tag{6.7}$$

, where

$|\,.\,| =$ total number or cardinality,

$MG =$ machine-group matrix that contains mg_k cells, where k is the index of machine-cells,

$k =$ 1, 2, …, $|MG|$,

$C_i =$ component, where i is the index of components, i = 1, 2, …, $|C|$,

$M_j =$ machine, where j is the index of machines, j = 1, 2, …, $|M|$,

$V_i =$ unit inter-cell material movement cost of component C_i,

$V_{ij} =$ unit inter-cell material movement cost of component C_i at machine M_j,

$D_i =$ demand of component C_i,

$D_{ij} =$ demand of component C_i at machine M_j,

$N_i =$ number of inter-cell material movements of component C_i, where

$$N_i = \sum_{k=1}^{|MG|} \sum_{j=1}^{|M|} o_{ij} \times (1 - q_{jk}) \tag{6.8}$$

N_j = number of inter-cell material movements at machine M_j, where

$$N_j = \sum_{k=1}^{|MG|} \sum_{i=1}^{|C|} o_{ij} \times (1 - q_{jk}) \tag{6.9}$$

E_i = unit intra-cell void element cost of component C_i,

E_{ij} = unit intra-cell void element cost of component C_i at machine M_j,

G_i = number of void elements of component C_i, where

$$G_i = \sum_{k=1}^{|MG|} \sum_{j=1}^{|M|} (1 - o_{ij}) \times q_{jk} \tag{6.10}$$

G_j = number of void elements at machine M_j, where

$$G_j = \sum_{k=1}^{|MG|} \sum_{i=1}^{|C|} (1 - o_{ij}) \times q_{jk} \tag{6.11}$$

w_N = weight of inter-cell material movement cost

w_G = weight of intra-cell cost of void elements,

$$w_N + w_G = 1 \tag{6.12}$$

$$o_{ij} = \begin{cases} 1, & \text{when component } C_i \text{ is produced on machine } M_j \\ 0, & \text{otherwise} \end{cases}, \text{ and}$$

$$q_{jk} = \begin{cases} 1, & \text{when machine } M_j \text{ is assigned to machine-group } mg_k \\ 0, & \text{otherwise} \end{cases}.$$

The $F(MG)$, $F(C_i)$, and $F(M_j)$ functions, as shown in equations (6.5) to (6.7), represent costs that are incurred when we form machine-cells. The $F(MG)$ is the function to calculate the total cost of the entire MG matrix. Equations (6.6) and (6.7) calculate the inter-cell and intra-cell material movement costs for only a portion of the MG matrix. In

equation (6.6), $F(C_i)$ calculates the costs with respect to the cell, mg_k. It sums up the costs incurred for each machine (a row in the m/c matrix) if it is assigned to the cell, mg_k, where $|M_{mg_k}|$ is the total number of machines in the mg_k cell. In equation (6.7), $F(M_j)$ computes the costs similar to equation (6.6). The difference is that equation (6.7) adds up the costs for each component (a column in the m/c matrix) which assigns to the cell, mg_k, where $|C_{mg_k}|$ is the total number of components in the mg_k cell.

The aforementioned functions are composed of two terms, the inter-cell material movement cost and the intra-cell cost of void elements. The first terms of the *F* functions are the weighted summation of the inter-cell material movement cost (*VDN*). The inter-cell material movement cost is often considered as an important measurement to evaluate a cellular manufacturing system. The product demand (*D*) is considered in the computation in order to obtain a practical machine-group arrangement. To compute the inter-cell material movement (*N*), we apply equations (6.8) and (6.9) which captures inter-cell material movements, i.e. the non-zero elements outside the diagonal cells.

The second terms of the *F* functions indicate the weighted intra-cell cost of void elements (*G*) for all components C_i's. A void element is an empty or a zero element inside a diagonal cell. The density of each cell is considered as a significant indicator of the efficiency of a cell formation solution. The higher density a cell has the better cell formation. Therefore, minimization of the second term can improve machine utilization. Equations (6.10) and (6.11) calculate the number of void elements for the corresponding column (component) *i* and row (machine) *j* in the m/c matrix. For the experiments conducted in this research, we weigh both terms equally, $w_N = w_G = 0.5$.

Identifying machines to be grouped in a cell is an optimization problem. The DCM initializes machine-groups by employing algorithms proposed by Chen [154] that builds association rules from all pairs of machines, places machines into cells according to their highest support values, and places each part into cells based on the maximum number of operation between the part and the machines in the cell. To reduce the chance of obtaining local optima that usually associate with greedy algorithms [155], the DCM makes selections of a machine-component to be in a cell aiming to minimize overall F as well as to maximize the grouping efficacy by incorporating an idea of reevaluating a criterion function from the Sequential Forward Floating Selection (SFFS) algorithm [128]. The SFFS is a selection procedure that repeatedly includes or excludes features (machines and components) by evaluating a criterion function (F) when it forms a new set of features. By following the idea of SFFS, the DCM is able to iteratively adjust the machines and components in the currently formed machine-groups, through the following procedures – **AdjustCells**, **AdjustMachines**, and **AdjustComponents**, to improve the total cost. Figure 6.18 depicts a flowchart of the DCM algorithm.

The input parameters for the DCM algorithm (Figure 6.17) are as follows: C is a set of components, where $C_i \in C$ and i = 1, 2, …, |C|; M is a set of machines, where $M_j \in M$ and j = 1, 2, …, |M|; D is a demand vector, where $D_l \in D$ and l = 1, 2, …, |D|; MC is a machine-component matrix; BOM is a matrix that represents BOM structure; V is a unit inter-cell material movement cost matrix, which V will incur costs to the total cost (F) only when N_i or N_j value is 1; E is an intra-cell void element cost matrix, which E will incur costs to F only when G_i or G_j value is 1 (equations (6.5) to (6.7) detail the cost

calculations); *max_m* is an integer indicates the maximum number of machines allowed for each cell.

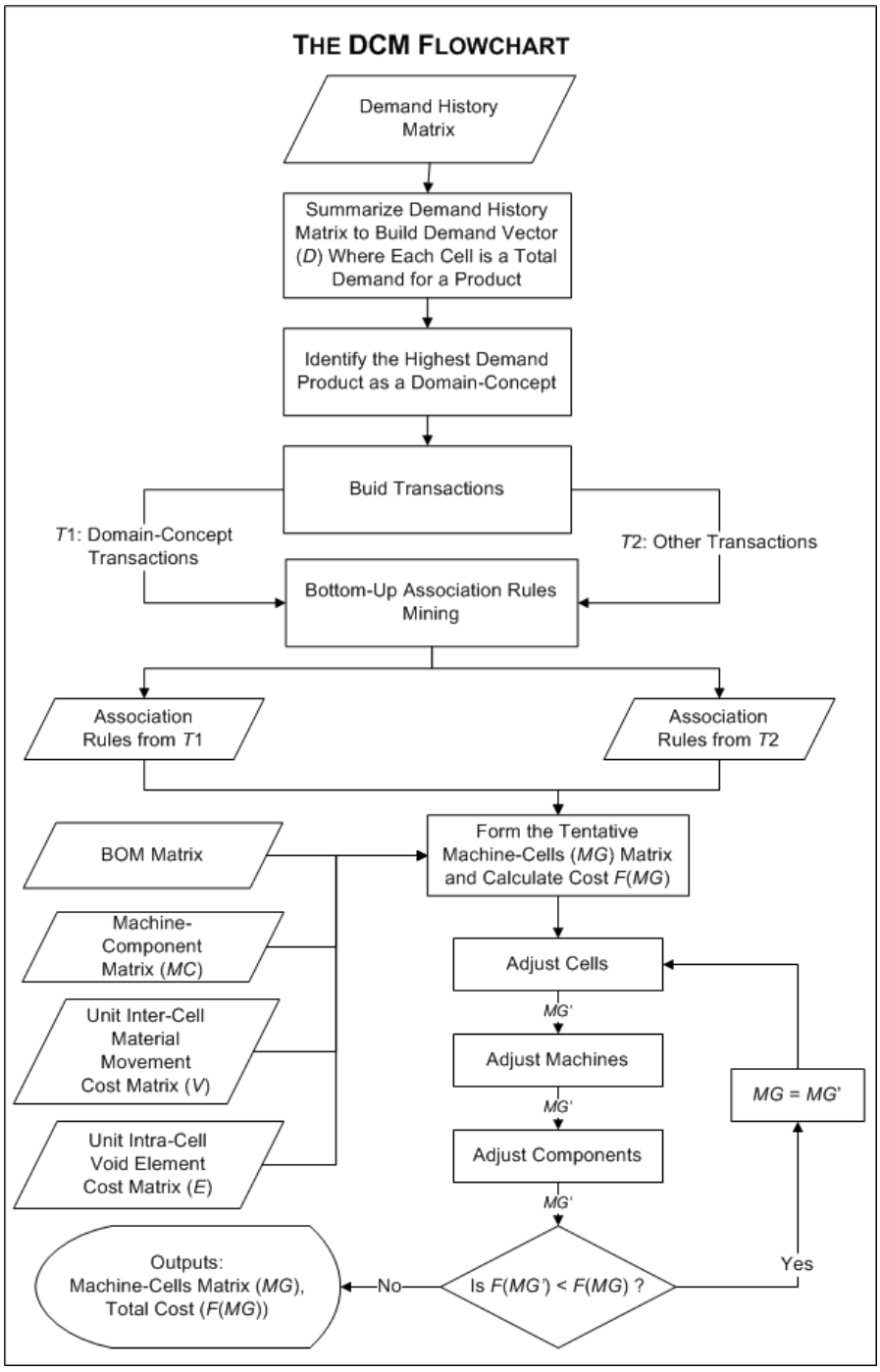


Figure 6.18. A flowchart of the DCM algorithm.

Let AR be a set of association rules; $AR_{T_{dc}}$ be a set of association rules from a transaction matrix T_{dc} where dc = 1 indicates domain-concept based transactions, and dc = 2 indicates other transactions (with a setting of two partitions); MG be a machine-group matrix that contains mg_k groups, and k = 1, 2, … , $|MG|$. The outputs of the DCM algorithm are MG and $F(MG)$.

At line 1 of the DCM algorithm in Figure 6.17, the x highest demand products, p_x, are identified as domain-concepts. For each p_x, the DCM algorithm calls **BuildTransactions** to generate two transaction matrices, T_1 (the group of MC transactions that belong to the domain-concept p_x) and T_2 (the group of other MC transactions). Please note that a flexible domain-concept setting with any number of partitions could be set. Lines 2 to 18 of the **BuildTransactions** algorithm separates transactions that are derived from demands (D) to build T_1, where line 7 shows that each demand contains components $(C_i \in C)$ and their quantities (*quant*). The variable *quant* is be used at line 17 to update and build a non-binary MC matrix that reflects the quantity of each machine-component (MC_{ij}). At line 18, the **BuildTransactions** updates the unselected demands (D'). This D' is be used when the algorithm builds T_2 at lines 19 to 20 by performing steps 4 to 17 using M' (machines that are not associated with p_x) and C' (components that are not associated with p_x), and D'. The **BuildTransactions** terminates at line 21 and return T_1, T_2, and non-binary MC to the DCM.

After the results of the **BuildTransactions** algorithm have been returned, the DCM continues with association rules mining process at line 3. The DCM then extracts two sets of association rules, one for T_1, and another for T_2, where each association rule

contains two machines. From lines 4 to 6, the DCM forms a tentative machine-group (*MG*) by incorporating Chen's approach [154] that first places machines into cells based on the support values then places components into cells based on the number of operations between components and cells. Moreover, the DCM also maintains the following: 1. Placing machines that are from the set of rules from the domain-concept, $AR_{T_{dc}}$, before placing the rest of the machines, and 2. Arranging components that are from the same BOM sub-structure. At line 7, the DCM calculates the cost $F(MG)$ of its tentative cell formation *MG*. From lines 8 to 11, the algorithm iteratively processes the **AdjustCells** to obtain a stabilized cell formation. The DCM finally returns the set of the final machine-groups *MG* and the cost $F(MG)$ at line 12.

The **AdjustCells** is an evaluation process of machines, using $F(M_j)$ function through **AdjustMachines** sub-procedure, and components, using $F(C_i)$ function through **AdjustComponents** sub-procedure. The **AdjustMachines** and the **AdjustComponents** work similarly. The former is to reevaluate each machine and reassign the machine to the cell that incurs the minimum cost with a criterion that the size of the newly selected machine-group, $mg_{selected\_cell}$, is not more than the maximum number of machines allowed for a cell (as indicated by *max_m*). The latter is to reevaluate each component then reassign the component to the cell that incurs the minimum cost with a criterion that the component is from the same BOM sub-structure as the others in the cell. Both of the sub-procedures update and return *MG* to the **AdjustCells**. More detail explanations are as follows.

The **AdjustCells** procedure starts by executing its sub-procedure, **AdjustMachines**, to evaluate each machine, M_j, against the currently assigned

components, C_i's, of each machine-group, mg_k, in MG. In **AdjustMachines**, all machines are evaluated and to be reassigned to other cells if the cost $F(M_j)$ is reduced. The **AdjustMachines** returns an updated machine-group matrix to the **AdjustCells** which then compares whether the new machine-groups (MG') matrix is different from the previous machine-cells (MG) at line 2. If the cost is improved, the **AdjustCells** executes the **AdjustComponents** sub-procedure at line 3. This is to evaluate each component, C_i, against the machines, M_j's, of each machine-group, mg_k, in MG. However, if there is no improvement in cost, the **AdjustCells** terminates without further executing the **AdjustComponents**. The **AdjustCells** returns the original MG and the cost $F(MG)$ to the DCM at line 5.

6.5.4 Results, Analysis and Discussions

The experiments are conducted on a standard server with an Intel Xeon IV 2.40GHz CPU and 1 gigabytes memory machine. The DCM program and its modules are written in Java programming language (JDK 1.5). There are two experiments conducted to evaluate the DCM algorithm. The first experiment is to demonstrate the DCM algorithm is able to produce comparable results to existing methods on binary data set using a single domain setting without constraints. This experiment is conducted on 20 data sets to demonstrate the grouping efficacy. On average, the computation time for this data collection is approximately 0.153 seconds.

The second experiment is conducted using a randomly generated data set of the m/c matrix with dimensions of 200 machines and 2,000 components, where each value in the matrix represents a multiplication value of $MC_{ij} * V_i$ that can be any positive number

rather than 0 or 1. This data set also includes other input parameters. They are a product demand vector (*D*) and a BOM matrix that is generated with two criteria, a maximum fan-out of 20 and a maximum height of 6 for each component path. Due to space limitations, only a subset of the second collection with the dimensions of 25 machines and 14 components that utilizes the BOM structure from Figure 6.16 and Table 6.14 are shown in this section. The average computation time from the experiments using the second data collection is about 6 minutes.

The grouping efficacy measure, Γ, introduced by Kumar and Chandrasekharan [156] is used to evaluate the experimental results of the proposed DCM algorithm and to compare with other approaches. The Γ formula is as follow.

$$\Gamma = 1 - \frac{e_v + e_0}{e + e_v} = \frac{e - e_0}{e + e_v} \tag{6.13}$$

, where e is the total number of non-zero cells in the matrix, e_v is the total number of zero cells inside machine-groups, meaning that there is no component produced by the particular machine, and e_0 is the total number of non-zero cells outside the machine-groups, meaning that the component has to be transported among machine-groups. An ideal grouping result has Γ =1

Table 6.16 shows the grouping efficacy (Γ) values from the experiments conducted using the known binary m/c matrices, but without constraints, such as BOM and product demands. The DCM algorithm takes a binary m/c matrix and a maximum number of machines allowed for each cell as its input parameters. The algorithm reports Γ values of the tentative cell formations, the iterations, and the final cell formations. An iteration involves a series of machine movement and component rearrangement.

Moreover, Table 6.17 details the Γ values of the same experiments as above by comparing the DCM with other approaches. The DCM approach has comparable results to the ARI and the GP-SLCA. However, either the ARI or the GP-SLCA can provide flexible mechanism to take into considerations of BOM and demands.

Table 6.16. The Details of the DCM Experiments Using Known Binary m/c Matrices

No	Data set	Size	*e*	Tentative *MG* Γ	Iteration 1 Γ	Iteration 2 Γ	Final *MG* Γ	Final *MG*'s characteristics		Time (milli-seconds)
								Number of machines in a cell (at most)	Number of cells	
1	Boctor 1[157]	16x30	121	0.457	0.492	n/a	0.492	8	3	100
2	Boctor 2 [157]	16x30	106	0.571	0.579	0.609	0.609	6	3	100
3	Boctor 3 [157]	16x30	92	0.583	0.7	n/a	0.700	5	4	140
4	Boctor 4 [157]	16x30	111	0.411	0.462	n/a	0.462	7	4	130
5	Boctor 5 [157]	16x30	107	0.658	0.727	n/a	0.727	5	4	130
6	Boctor 6 [157]	16x30	101	0.533	0.766	n/a	0.766	5	4	140
7	Boctor 7 [157]	16x30	112	0.593	0.732	n/a	0.732	7	4	101
8	Boctor 8 [157]	16x30	114	0.562	0.579	n/a	0.579	7	5	100
9	Boctor 9 [157]	16x30	118	0.595	0.774	n/a	0.774	6	4	101
10	Boctor 10 [157]	16x30	108	0.593	0.638	n/a	0.638	5	5	140
11	Boe and Cheng [158]	20x35	153	0.474	0.556	n/a	0.556	6	4	150
12	Burbidge [159]	16x43	126	0.544	0.561	n/a	0.561	4	6	130
13	Carrie [160]	20x35	136	0.475	0.757	n/a	0.757	5	4	190
14	Chandrasekharan and Rajagopalan [146]	40x100	420	0.476	0.840	n/a	0.840	6	10	461
15	Chandrasekharan and Rajagopalan [145]	8x20	61	0.656	0.852	n/a	0.852	4	3	51
16	Chandrasekharan and Rajagopalan 1 [161]	24x40	131	0.611	1.000	n/a	1.000	5	7	180
17	Chandrasekharan and Rajagopalan 2 [161]	24x40	130	0.833	0.851	n/a	0.851	5	7	251

No	Data set	Size	*e*	Tentative *MG* Γ	Iteration 1 Γ	Iteration 2 Γ	Final *MG* Γ	Final *MG*'s characteristics		Time (milli-seconds)
								Number of machines in a cell (at most)	Number of cells	
18	Chandrasekharan and Rajagopalan 3 [161]	24x40	131	0.551	0.677	0.735	0.735	5	7	210
19	Chandrasekharan and Rajagopalan 5 [161]	24x40	131	0.373	0.428	0.455	0.455	2	2	201
20	Seifoddini [162]	11x12	78	0.646	0.731	n/a	0.731	4	3	60

Table 6.17. The Grouping Efficacy (Γ) Values as the Experimental Results Comparisons among Various Approaches Using Known Binary m/c Matrices

No	DCM	ARI	GP-SLCA	ZODIAC	GRAFICS	MST-GRAFICS	MST	GA-TSP	SLINK	ALINK
1	0.492	n/a	0.509	0.349	0.481	0.447	n/a	n/a	n/a	n/a
2	0.609	0.571	0.618	0.586	0.534	0.508	n/a	n/a	n/a	n/a
3	0.700	0.708	0.700 §	0.686	0.675	0.644	n/a	n/a	n/a	n/a
4	0.462	0.478	0.496	0.267	0.449	0.407	n/a	n/a	n/a	n/a
5	0.727	0.727	0.727 §	0.727	0.691	0.727	n/a	n/a	n/a	n/a
6	0.766	0.766	0.782	0.764	0.771	0.760	n/a	n/a	n/a	n/a
7	0.732	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
8	0.579	0.579	0.774 §	0.320	0.579	0.530	n/a	n/a	n/a	n/a
9	0.774	0.774	n/a	0.774	0.774	n/a	n/a	n/a	n/a	n/a
10	0.638	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
11	0.556	0.527	0.568	0.511	n/a	0.471	n/a	0.551	n/a	n/a
12	0.561	0.549	0.568 §	0.538	0.544	n/a	n/a	0.539	0.544	0.483
13	0.757	0.751	0.767	0.751	0.751	n/a	n/a	0.753	n/a	n/a
14	0.840	0.842	0.840 §	0.839	0.839	n/a	0.831	0.840	n/a	n/a
15	0.852	0.852	0.852 §	0.852	0.852	n/a	0.852	0.852	n/a	n/a
16	1.000	1.000	1.000 §	1.000	1.000	n/a	1.000	1.000	n/a	n/a
17	0.851	0.851	0.851 §	0.851	0.851	n/a	0.851	0.851	n/a	n/a
18	0.735	0.735	0.735 §	0.730	0.735	n/a	0.730	n/a	n/a	n/a
19	0.455	0.520	0.479	0.204	0.433	0.466	n/a	0.494	n/a	n/a
20	0.731	0.742	0.731 §	0.731	0.731	n/a	n/a	n/a	0.522	0.720

§ denotes the resulting machine-groups contain one or more singletons, where a singleton is a machine-cell that has only one machine.

Table 6.18 shows the resulting Γ values and the *F* values of machine-group matrices from three settings when apply the DCM algorithm with the subset of the data in the second experiment. Each setting has a different maximal number of machines per cell. For all settings, we execute the DCM with the complex constraints. Please note that the demand values (D_i) are randomly generated, and they associate with the BOM shown in Figure 6.16 and the parameters listed in Table 6.15. The resulting Γ values and the *F* values consistently agree, e.g. the setting that results in a better Γ value (higher) also has the better *F* value (lower).

Table 6.18. The Resulting Grouping Efficacy (Γ) and Total Cost (*F*) as Measurement Values when the DCM Algorithm Is Applied to the Subset of 200x2000 Data Set in Various Machine-Group Settings

Measurement Values	**2 machine-groups with *max_m* = 13**	**3 machine-groups with *max_m* = 11**	**4 machine-groups with *max_m* = 9**
Γ	0.192	0.326	0.347
F	86,275.5	54,506.5	50,390.5

As shown in Table 6.18, the DCM generates the *MG* matrix with three machine-cells that optimizes the total cost (*F*) under the *max_m* constraint of 11. The *F* value calculated from the DCM is 54,504.5. The Γ value obtained from the experiment with this setting is 0.326. However, it is important to mention that the low Γ value is a result of four factors: 1.) fully applying the DCM algorithm with the objectives of rearranging m/c matrix in favor of the highest demand products as domain-concepts, 2.) rearranging m/c matrix by the DCM with a restriction of the given BOM, 3.) allowing only m/c's that minimize the cost (*F*) being added into cells, and 4.) the sparseness of the generated data set is high.

Table 6.19. *MG* Grouping Experimental Results with Three Groups of Machines (*max_m* = 11) from an m/c Matrix of Size 25x14, a Subset of the 200x2000 Data Set

m/c	*P3*	*Pb*	*P2*	*P6*	*P4*	*P11*	*Pa*	*P8*	*P1*	*P10*	*P9*	*Pc*	*P7*	*P5*
M2	5900	875												
M13	5900	875												
M11			1729	1750	70									
M25				1750	70					480				
M18				1750	70									
M15				1750					567					
M6				1750										
M5		875												
M8			1729					210						
M1					70	2400	270				90			
M7					70	2400								
M16						2400					90			
M24		875				2400					90			
M12			1729			2400		210						
M19							270	210		480				
M4							270	210	567					
M9										480	90	150		
M10							270		567					
M21									567					
M14										480		150	75	420
M17						2400						150	75	
M3			1729							480		150		
M22					70		270	210					75	420
M23	5900								567					420
M20			1729											420
V_i	5900	875	1729	1750	70	2400	270	210	567	480	90	150	75	420
N_i	1	1	3	0	3	1	1	2	2	3	0	1	0	0
E_i	5900	875	1729	1750	70	2400	270	210	567	480	90	150	75	420
G_i	7	6	7	4	6	5	6	7	7	8	6	3	3	2
F_i	23600	3062.5	8645	3500	315	7200	945	945	2551.5	2640	270	300	112.5	420

As shown in Table 6.19, the last five rows show the resulting unit inter-cell material movement costs (V_i), number of inter-cell material movements of component *i*

(N_i), unit intra-Cell void element costs (E_i), number of void elements of component i (G_i), and total costs of each component (F_i). The total cost of the entire *MG* (F) is 54,506.5.

Table 6.20. *MG* Grouping Experimental Results with Four Groups of Machines (*max_m* = 9) Using the Same Data Set as Table 6.19

m/c	*P3*	*Pb*	*P2*	*P6*	*P4*	*P11*	*Pa*	*P8*	*P1*	*P10*	*Pc*	*P9*	*P7*	*P5*
M2	5900	875												
M13	5900	875												
M11			1729	1750	70									
M25				1750	70					480				
M18				1750	70									
M15				1750					567					
M6				1750										
M5		875												
M8			1729					210						
M1					70	2400	270					90		
M7					70	2400								
M16						2400						90		
M24		875				2400						90		
M12			1729			2400		210						
M19							270	210		480				
M4							270	210	567					
M10							270		567					
M21									567					
M9										480	150	90		
M14										480	150		75	420
M3			1729							480	150			
M17						2400					150		75	
M22					70		270	210					75	420
M23	5900								567					420
M20			1729											420
V_i	5900	875	1729	1750	70	2400	270	210	567	480	150	90	75	420
N_i	1	1	3	0	3	1	1	2	2	2	0	3	2	1
E_i	5900	875	1729	1750	70	2400	270	210	567	480	150	90	75	420
G_i	7	6	7	4	6	4	5	6	6	1	0	3	2	0
F_i	23600	3062.5	8645	3500	315	6000	810	840	2268	720	0	270	150	210

Table 6.20 is used to directly compare the *MG* results with Table 6.19. DCM generates the *MG* matrix result with four machine-cells that optimizes the total cost (*F*) under the *max_m* constraint of 9. The experimental results shown in this table give an *F* value of 50,390.5 and the Γ value of 0.347, where both values indicate that the resulting *MG* matrix in Table 6.20 is better than the one in Table 6.19. If there is no *max_m* constraint, the DCM algorithm will execute all possible settings ($2 \leq max\_m \leq |M|$) and select the lowest cost for the final formation.

In conclusion, from both collections of the experiments discussed in this section, the DCM algorithm has demonstrated that it is not only able to generate comparable grouping efficacy results when applied to the documented data sets, but also possesses the advantages of flexibility, efficiency, and applicability for large-scale and complex cellular manufacturing settings that can optimize costs while maintaining the production requirements based on a given BOM.

6.5.5 Conclusion and Future Work

The formation of cellular manufacturing is an indispensable procedure for the implementation of flexible manufacturing systems. The proposed DCM algorithm provides an effective method for such a task. To solve the large-scale and complex cell formation problems, DCM applies an AR approach with a consideration of real-world factors, which include the machine-component relationships, the demands for the products, the inter-cell material movement costs, and the intra-cell void element costs. DCM forms manufacturing cells by grouping the machines and parts according to their

associations and relationships, while also balancing the possible highest interaction within cells and the lowest inter-cell movements.

From the experimental results, we found that the DCM algorithm could be used to solve various cell formation problems with results that are at least as efficient as other approaches in terms of grouping efficacy values using binary m/c matrices. Moreover, the DCM algorithm has its main advantages over the other approaches which include the following:

1.) The ability to handle more parameters rather than the co-occurrence of machine-component in term of binary matrices,
2.) Its ability to efficiently handle bigger data set sizes,
3.) Its capability to optimize machine-groups according to the criterion function while making decisions of adding a machine into a group, and
4.) The ability to allow machine-cells to be reevaluated by a series of inclusion and exclusion processes to improve a preset criterion value.

Cell rearrangement may be required in various manufacturing situations, such as machine breakdown and machine/part inclusion after the machine-group matrix has been determined. In practice, the DCM algorithm has the capability of dealing with cell rearrangement because it can regenerate the cell formation speedily. System managers thus can use the information to make an appropriate decision. Moreover, since the DCM algorithm includes an extensive set of manufacturing parameters, the resulting cell formation is more down-to-earth.

Further improvements to this ongoing research include the following: 1.) Assistance in decision-makers definition and validation of Bill of Material because there are currently no limitations of PART-OF relationship structures, 2.) Prioritizing and weighting the influences of BOM, demands, the inter-cell movement costs, and the intra-cell void element costs in the DCM algorithm, 3.) Adding the ability to give an incentive when grouping machine-cells based on the BOM and to assign penalty otherwise, and 4.) Adding the ability to collect the production information as feedback and to use this information to further improve machine groupings.

CHAPTER 7

CONCLUSION AND FUTURE WORK

The Domain-Concept Mining (DCM) approach has been empirically proven useful to a vast range of data sets, including but not limited to public health and biomedical informatics data, as demonstrated in the previous chapters. This chapter will serve as a conclusion and discussion of the challenging research problems yet unsolved, which are the on-going and future work of this research.

7.1 Conclusion

As a comparison between the traditional brute-force data mining approaches and DCM, a summary of the common problems from the traditional approaches (such as Apriori [13] and FPT [20]) that DCM has overcome are listed in Table 7.1.

The main reason that DCM can solve the listed problems is because it was originally designed to respond to human experts' preferences, needs, and expectations. The approach starts by utilizing the experts' valuable experiences in organizing data before analysis. Further, DCM takes into account the findings that are expected by the experts to be part of the uncovered associations. This is done by partitioning the data in a way that DCM can directly mine associations from dc partitions, which are (*attribute*: *value*) pairs representing the valuable findings.

Table 7.1. Comparisons between Traditional Mining and DCM

Traditional Brute-Force Mining Issues	DCM
Memory exhaustion with large data sets and/or a low minimum support threshold.	Multiple domain-concept partitions (where each is magnitude smaller than the original data) that can be independently mined in batch, distributed, or parallel fashion with much less computer resource requirements.
Unorganized results.	Results are organized according to their natural distributions, i.e. their domain-concept partitions.
Minimum support threshold is neither sensitive nor specific.	One global minimum support threshold is needed for all domain-concept partitions. The threshold is automatically adjusted according to each domain-concept's distribution. Hence, the threshold is sensitive and specific.
Valuable items with low probabilities cannot be uncovered, not efficiently uncovered, or are buried in an overwhelming amount of unorganized results.	Valuable results can be domain-concept partitions. As a result, DCM can specifically mine these partitions; hence, DCM needs less computational resources. Moreover, the associations uncovered from these partitions are directly related to the valuable results, and they are organized.
Mine data with an assumption of no prior hypothesis.	DCM granulizes partitioning criteria and groups related transactions together. DCM adds no other prior hypothesis to domain-concepts; hence, the mining results are completely data driven and no bias.
The number of items in the longest association is too few; hence, the association represents only a limited co-occurrences of the (*attribute*: *value*) pairs.	With DCM-PA, the longest association contains many more items than the traditional approach. In addition, there are many more of such valuable (different) associations offered to human experts.
Cannot be directly implemented to temporal, spatial, or especially incremental data sets.	DCM is an approach that is designed to partition then mines the data; hence, it is a good fit with temporal, spatial, and incremental data sets. DCM-PA intelligently aggregates DCM's offline results from various partitions through pipelining technique. Therefore, these results are not required to be materialized. Further, DCMiner gains advantages from these special data sets by being able to offer various visualization formats (including trends) that fit to the nature of the data.
Estimation and heuristic approaches often used to improve the traditional approaches.	DCM and DCM-PA are brute-force; hence, the results are the complete set of frequent itemsets.

Moreover, dc partitions are unique because their sizes are different, and they may overlap each other. However, these dc partitions' properties are DCM's advantages, not drawbacks, because dc partitions represent the distribution of the data. Hence, the *global* minimum support threshold is automatically adjusted once used for mining each dc partition, accordingly. This allows DCM to be sensitive (uncover valuable findings), and specific (uncover valuable findings directly while using less computational resources).

Moreover, the DCM-PA approach, which is based on Bayes Theorem [31], can intelligently and correctly aggregate various sizes partitions that may overlap each other. DCM-PA offers the flexibility to the human experts to adjust the results on-demand and online by being more specific (intersect dc partitions) or more broad (union dc partitions). These aggregation abilities increase the understandability and usability of the uncovered results.

In conclusion, the combination of DCM, DCM-PA, and DCMiner makes it possible to offer an efficient on-demand, and yet brute-force, data mining approach for the human experts to fully utilize and benefit from their large data sets.

7.2 Future Work

This section details challenging research topics, on-going, and future work of Domain-Concept Mining (DCM) and DCM Partition Aggregation (DCM-PA) on temporal, spatial, and incremental data sets, and DCM Web system (DCMiner).

Challenging research topics for DCM and DCM-PA include:

1.) Association mining with partition when items' values are continuous. Currently, DCM categorized all continuous attributes by utilizing well-accepted scales from the CDC's BRFSS definitions. However, there can be other continuous values beyond those of health-related. Therefore, this future work is to explore approaches, which include techniques to discretize continuous values using equi-width (equal ranges), equi-depth (equal number of transactions among the bins), and other statistical analyses. Particularly for the statistical analysis, which has been briefly explored in the DCM's work for NIS 2005 data set, it is to bin the continuous values according to their distributions around the average values.

2.) Following the above topic, a future work also includes an exploration of "crisp" as compared to "fuzzy" values of attributes. This is because not all attributes' values can be categorized as "yes" or "no", "male" or "female", etc. Uncertainty and degree of truth should also be considered when binning continuous values. There can also be overlapping areas among the neighboring bins.

3.) Association mining that takes into consideration of quantities. Using a market basket analysis as an example, two loafs of white bread in a basket should not be considered only as the bread item exists in the transaction. A potential approach is to assign different weights to represent different quantities.

4.) Association mining that is suitable for multiple-choice multiple-answer surveys. So far, DCM assumes that each transaction represents a survey (or a person), and each question is answered once for each survey. This assumption

is not applicable to many other real-world data sets, such as the lymphedema (one patient with multiple visits) and other general questionnaire-type data.

5.) Explorations of association ranking in order to utilize the associations as predictors. Currently, DCM mining results may be repeated according to their actual distributions and associations; hence, DCM is a "descriptive" association mining approach. For example, the (*diabetes*: *yes*) item can associate with any number of domain-concepts. The future work is to explore which domain-concept(s) is/are best for an item to be associated with. Hence, this work would enable associations to have a strong predictive power, i.e. the associations may be used to predict domain-concepts when new or incremental transactions are submitted to the system.

6.) Explorations of a root cause and a solution of how to best uncover associations from the under-represented groups of a population. The current explorations using correlations and hybrid thresholds are still not sufficient enough in identifying these valuable associations.

7.) Enabling a "complete" on-demand mining, i.e. allow newly collected data to be submitted on-line for DCM to partition and then mine efficiently. Subsequently, DCMiner and DCM-PA include these new results into the online systems on-the-fly.

The following details the on-going and future work of DCMiner. A goal of the DCMiner Web-based interface is to serve as an incremental data mining tool, where human experts can upload a series of newly collected data, as they become available, for DCM to mine for frequent itemsets. After that, DCM-PA incrementally aggregates

(union) the new and previous sets of results (frequent itemsets). Finally, DCMiner offers: 1.) a result viewing tool that can compare and contrast between a series of results, 2.) an aggregation tool that the human experts can flexibly merge (union or intersection) these results according to their domain-concept selections. Particularly for the DCMiner result viewing tool, visualization techniques, such as line graphs (for trends), histograms, and pie charts can be highly suitable as representations of the frequent itemsets and their temporal patterns. Lastly, for data sets such as the NIS and the BRFSS, with spatial dimensions from data collected from states, (hospital) region, metropolitan and rural areas, etc., "Google Earth" [93] will be integrated to increase the understandability of the results.

APPENDIX A

BRFSS 2006 SELECTED VARIABLES

Table Appendix A.1. BRFSS 2006 Selected Variables

itemID	Variable Name	Description	Variable Value	Variable Meaning	Interesting Indicator	Frequency	Percentage
1	*GENHLTH*	Would you say that in general your health is:	*1*	Excellent	N	67337	18.93
2	*GENHLTH*	Would you say that in general your health is:	*2*	Very good	N	113348	31.87
3	*GENHLTH*	Would you say that in general your health is:	*3*	Good	N	107288	30.16
4	*GENHLTH*	Would you say that in general your health is:	*4*	Fair	Y	46373	13.04
5	*GENHLTH*	Would you say that in general your health is:	*5*	Poor	Y	20005	5.62
6	*HLTHPLAN*	Do you have any kind of health care coverage, including health insurance, prepaid plans such as HMOs, or government plans such as Medicare?	*1*	Yes	N	313248	88.06
7	*HLTHPLAN*	Do you have any kind of health care coverage, including health insurance, prepaid plans such as HMOs, or government plans such as Medicare?	*2*	No	Y	41492	11.66

itemID	Variable Name	Description	Variable Value	Variable Meaning	Interesting Indicator	Frequency	Percentage
8	*MEDCOST*	Was there a time in the past 12 months when you needed to see a doctor but could not because of cost?	*1*	Yes	N	41035	11.54
9	*MEDCOST*	Was there a time in the past 12 months when you needed to see a doctor but could not because of cost?	*2*	No	Y	313853	88.23
10	*DIABETE2*	Have you ever been told by a doctor that you have diabetes?	*1*	Yes	Y	36085	10.14
11	*DIABETE2*	Have you ever been told by a doctor that you have diabetes?	*3*	No	N	311704	87.63
12	*CVDINFR*	Has a doctor, nurse, or other health professional ever told you that you had a heart attack (a myocardial infarction)?	*1*	Yes	Y	20354	5.72
13	*CVDINFR*	Has a doctor, nurse, or other health professional ever told you that you had a heart attack (a myocardial infarction)?	*2*	No	N	333726	93.82
14	*CVDSTRK*	Has a doctor, nurse, or other health professional ever told you that you had a stroke?	*1*	Yes	Y	13150	3.7
15	*CVDSTRK*	Has a doctor, nurse, or other health professional ever told you that you had a stroke?	*2*	No	N	341643	96.06
16	*ASTHMA*	Have you ever been told by a doctor, nurse, or other health professional that you had asthma?	*1*	Yes	Y	45843	12.89
17	*ASTHMA*	Have you ever been told by a doctor, nurse, or other health professional that you had asthma?	*2*	No	N	308931	86.86
18	*QLACTLM*	Are you limited in any way in any activities because of physical, mental, or emotional problems?	*1*	Yes	Y	86460	24.31

itemID	Variable Name	Description	Variable Value	Variable Meaning	Interesting Indicator	Frequency	Percentage
19	*QLACTLM*	Are you limited in any way in any activities because of physical, mental, or emotional problems?	*2*	No	N	267758	75.27
20	*USEEQUIP*	Do you now have any health problem that requires you to use special equipment, such as a cane, a wheelchair, a special bed, or a special telephone? (Include occasional use or use in certain circumstances.)	*1*	Yes	Y	33069	9.3
21	*USEEQUIP*	Do you now have any health problem that requires you to use special equipment, such as a cane, a wheelchair, a special bed, or a special telephone? (Include occasional use or use in certain circumstances.)	*2*	No	N	322396	90.63
22	*MARITAL*	Are you married, divorced, widowed, separated, never been married, or a member of an unmarried couple?	*1*	Married	Y	196341	55.2
23	*MARITAL*	Are you married, divorced, widowed, separated, never been married, or a member of an unmarried couple?	*2*	Divorced	Y	49804	14
24	*MARITAL*	Are you married, divorced, widowed, separated, never been married, or a member of an unmarried couple?	*3*	Widowed	Y	45333	12.75
25	*MARITAL*	Are you married, divorced, widowed, separated, never been married, or a member of an unmarried couple?	*4*	Separated	Y	8339	2.34
26	*MARITAL*	Are you married, divorced, widowed, separated, never been married, or a member of an unmarried couple?	*5*	Never been married	Y	45430	12.77

itemID	Variable Name	Description	Variable Value	Variable Meaning	Interesting Indicator	Frequency	Percentage
27	*MARITAL*	Are you married, divorced, widowed, separated, never been married, or a member of an unmarried couple?	*6*	A member of an unmarried couple	Y	8970	2.52
28	*EDUCA*	What is the highest grade of year of school you completed?	*1*	Never attended school or only kindergarten	Y	630	0.18
29	*EDUCA*	What is the highest grade of year of school you completed?	*2*	Grades 1 through 8 (Elementary)	Y	12799	3.6
30	*EDUCA*	What is the highest grade of year of school you completed?	*3*	Grades 9 through 11 (Some high school)	Y	23983	6.74
31	*EDUCA*	What is the highest grade of year of school you completed?	*4*	Grade 12 or GED (High school graduate)	Y	107740	30.29
32	*EDUCA*	What is the highest grade of year of school you completed?	*5*	College 1 to 3 years (Some college or technical school)	Y	93399	26.26
33	*EDUCA*	What is the highest grade of year of school you completed?	*6*	College 4 years or more (College graduate)	Y	116169	32.66
34	*EMPLOY*	Are you currently employed for wages, self-employed, out of work, a homemaker, student, retired, or unable to work?	*1*	Employed for wages	Y	165724	46.6

itemID	Variable Name	Description	Variable Value	Variable Meaning	Interesting Indicator	Frequency	Percentage
35	*EMPLOY*	Are you currently employed for wages, self-employed, out of work, a homemaker, student, retired, or unable to work?	*2*	Self-employed	Y	31954	8.99
36	*EMPLOY*	Are you currently employed for wages, self-employed, out of work, a homemaker, student, retired, or unable to work?	*3*	Out of work for more than 1 year	Y	6158	1.73
37	*EMPLOY*	Are you currently employed for wages, self-employed, out of work, a homemaker, student, retired, or unable to work?	*4*	Out of work for less than 1 year	Y	7539	2.12
38	*EMPLOY*	Are you currently employed for wages, self-employed, out of work, a homemaker, student, retired, or unable to work?	*5*	Homemaker	Y	28542	8.03
39	*EMPLOY*	Are you currently employed for wages, self-employed, out of work, a homemaker, student, retired, or unable to work?	*6*	Student	Y	6832	1.92
40	*EMPLOY*	Are you currently employed for wages, self-employed, out of work, a homemaker, student, retired, or unable to work?	*7*	Retired	Y	85595	24.07
41	*EMPLOY*	Are you currently employed for wages, self-employed, out of work, a homemaker, student, retired, or unable to work?	*8*	Unable to work	Y	22200	6.24
42	*INCOME*	What is your annual household income from all sources?	*1*	Less than $10,000	Y	17895	5.03
43	*INCOME*	What is your annual household income from all sources?	*2*	$10,000 to $14,999	Y	19136	5.38
44	*INCOME*	What is your annual household income from all sources?	*3*	$15,000 to $19,999	Y	23973	6.74

itemID	Variable Name	Description	Variable Value	Variable Meaning	Interesting Indicator	Frequency	Percentage
45	*INCOME*	What is your annual household income from all sources?	*4*	$20,000 to $24,999	Y	29388	8.26
46	*INCOME*	What is your annual household income from all sources?	*5*	$25,000 to $34,999	Y	39964	11.24
47	*INCOME*	What is your annual household income from all sources?	*6*	$35,000 to $49,999	Y	50221	14.12
48	*INCOME*	What is your annual household income from all sources?	*7*	$50,000 to $74,999	Y	51837	14.58
49	*INCOME*	What is your annual household income from all sources?	*8*	$75,000 or More	Y	73370	20.63
50	*SEX*	Indicate sex of respondent	*1*	Male	Y	135408	38.07
51	*SEX*	Indicate sex of respondent	*2*	Female	Y	220302	61.93
52	*PROSTATE*	Have you ever been told by a doctor, nurse, or other health professional that you had prostate cancer?	*1*	Yes	Y	63103	63.15
53	*PROSTATE*	Have you ever been told by a doctor, nurse, or other health professional that you had prostate cancer?	*2*	No	N	32227	32.25
54	*EMTSUPRT*	How often do you get the social and emotional support you need?	*1*	Always	N	167954	48.82
55	*EMTSUPRT*	How often do you get the social and emotional support you need?	*2*	Usually	N	101052	29.37
56	*EMTSUPRT*	How often do you get the social and emotional support you need?	*3*	Sometimes	Y	39152	11.38
57	*EMTSUPRT*	How often do you get the social and emotional support you need?	*4*	Rarely	Y	11324	3.29

itemID	Variable Name	Description	Variable Value	Variable Meaning	Interesting Indicator	Frequency	Percentage
58	*EMTSUPRT*	How often do you get the social and emotional support you need?	*5*	Never	Y	17617	5.12
59	*LSATISFY*	In general, how satisfied are you with your life?	*1*	Very satisfied	N	156739	45.58
60	*LSATISFY*	In general, how satisfied are you with your life?	*2*	Satisfied	N	166330	48.37
61	*LSATISFY*	In general, how satisfied are you with your life?	*3*	Dissatisfied	Y	14415	4.19
62	*LSATISFY*	In general, how satisfied are you with your life?	*4*	Very dissatisfied	Y	3656	1.06
63	*INSULIN*	Are you now taking insulin?	*1*	Yes	Y	7922	26.43
64	*INSULIN*	Are you now taking insulin?	*2*	No	N	22014	73.45
65	*DIABPILL*	Are you now taking diabetes pills?	*1*	Yes	Y	21362	71.28
66	*DIABPILL*	Are you now taking diabetes pills?	*2*	No	N	8532	28.47
67	*FEETSORE*	Have you ever had any sores or irritations on your feet that took more than four weeks to heal?	*1*	Yes	Y	3368	11.24
68	*FEETSORE*	Have you ever had any sores or irritations on your feet that took more than four weeks to heal?	*2*	No	N	26468	88.33
69	*DIABEYE*	Has a doctor ever told you that diabetes has affected your eyes or that you had retinopathy?	*1*	Yes	Y	6314	21.08
70	*DIABEYE*	Has a doctor ever told you that diabetes has affected your eyes or that you had retinopathy?	*2*	No	N	23219	77.51
71	*AGEG*	Age groups	*1*	Age 18 to 24	Y	14	4.07
72	*AGEG*	Age groups	*2*	Age 25 to 34	Y	38	10.78
73	*AGEG*	Age groups	*3*	Age 35 to 44	Y	60	16.87
74	*AGEG*	Age groups	*4*	Age 45 to 54	Y	75	21.29
75	*AGEG*	Age groups	*5*	Age 55 to 64	Y	69	19.47

itemID	Variable Name	Description	Variable Value	Variable Meaning	Interesting Indicator	Frequency	Percentage
76	*AGEG*	Age groups	*6*	Age 65 to 74	Y	47	13.41
77	*AGEG*	Age groups	*7*	Age 75 or older	Y	41	11.56
78	*TOTINDA*	Adults that report doing physical activity or exercise during the past 30 days other than their regular job	*1*	Had physical activity or exercise	N	263	74.21
79	*TOTINDA*	Adults that report doing physical activity or exercise during the past 30 days other than their regular job	*2*	No physical activity or exercise in last 30 days	Y	91	25.68
80	*LTASTHM*	Adults who have ever been told they have asthma	*1*	No	N	308	86.85
81	*LTASTHM*	Adults who have ever been told they have asthma	*2*	Yes	Y	45	12.89
82	*BMICAT*	Three-categories of Body Mass Index (BMI)	*1*	Neither overweight nor obese (BMI less than 25.0)	Y	126	35.62
83	*BMICAT*	Three-categories of Body Mass Index (BMI)	*2*	Overweight (25.0 <= BMI <=29.9)	Y	123	34.71
84	*BMICAT*	Three-categories of Body Mass Index (BMI)	*3*	Obese (BMI 30.0 or greater)	Y	88	24.75

APPENDIX B

DCM-PA EXPERIMENTAL RESULTS

The summary of the experimental results from Domain-Concept Partition Aggregation (DCM-PA) that aggregates (*DIABETE2*: *1*) or itemID 10 (see Table Appendix A.1 for itemIDs and their meanings) with the other 83 domain-concepts of the "Centers for Disease Control and Prevention" (CDC)'s "Behavioral Risk Factor Surveillance System" (BRFSS) 2006 data set [33] is shown in Table Appendix B.2. The purpose is to demonstrate that DCM with DCM-PA is able to uncover many more valuable findings than the traditional brute-force association mining approach.

In total, there are 1,828 $(B \mid A_1 \vee A_2)$ aggregations, where A_1 is (*DIABETE2*: *1*), A_2 is one the possible 83 domain-concept partitions, and B is one of the other possible 82 itemIDs. Please note that it is not always the case that an aggregation is successful because there can be combinations of A_1, A_2, and/or B that do not actually exist in the data set, such as an aggregation between A_1 = (*DIABETE2*: *1*) and A_2 = (*EDUCA*: *1*). From this data set, the maximum number of A_2 to be aggregated with A_1 is 73, which is when B itemID is either 18 (*QLACTLM*: *1*) or 79 (*TOTINDA*: *1*).

From the BRFSS 2006 data set, (*DIABETE2*: *1*) domain-concept contains 36,085 transactions from 355,170 transactions in total. The last two columns in Table Appendix

B.1 are the minimum support values and the maximum support values in the form of $P(B \mid A_1 \vee A_2)$.

Table Appendix B.1. The Summary of the Results from DCM-PA Union Operations on the BRFSS 2006 Data Set with (*DIABETE2*: *yes*) as A_1

# of A_2 Domain-Concepts	**Domain-Concept *B* itemID**	**MIN** $P(B \mid A_1 \vee A_2)$	**MAX** $P(B \mid A_1 \vee A_2)$
62	4	0.12	0.31
32	5	0.13	0.23
24	12	0.12	0.19
70	16	0.11	0.61
73	18	0.19	0.56
40	20	0.12	0.29
69	22	0.39	0.71
62	23	0.11	0.20
51	24	0.11	0.38
29	30	0.10	0.13
69	31	0.21	0.38
69	32	0.23	0.29
67	33	0.18	0.49
66	34	0.20	0.52
60	40	0.20	0.60
31	41	0.11	0.21
24	44	0.10	0.12
34	45	0.10	0.12
55	46	0.11	0.14
56	47	0.12	0.16
43	48	0.10	0.18
72	50	0.29	0.82
71	51	0.53	0.76
56	56	0.11	0.19
6	63	0.22	0.22
20	65	0.20	0.59
6	69	0.17	0.17
64	74	0.16	0.27
65	75	0.18	0.29
55	76	0.13	0.33
40	77	0.11	0.35
73	79	0.20	0.44
70	81	0.11	0.61
72	82	0.15	0.39
72	83	0.29	0.42

APPENDIX C

PROOF DOMAIN-CONCEPT PARTITION AGGREGATION BY INDUCTION

This section is to illustrate a proof of the Domain-Concept Partition Aggregation (DCM-PA) approach by induction [164] based on the Inclusion-Exclusion Principle. The proof has been adapted from [165].

Theorem 1. The following statement is true for all $n \geq 1$.

$$\begin{aligned} \mathrm{P}\left(B \mid \left((\vee_{i=1}^{n-1} A_i) \vee A_n\right)\right) = \Big[& \sum_{i=1}^{n} \mathrm{P}\left(B \wedge A_i\right) \\ & - \sum_{1 \leq i < j \leq n} \mathrm{P}\left(B \wedge A_i \wedge A_j\right) \\ & + \sum_{1 \leq i < j < k \leq n} \mathrm{P}\left(B \wedge A_i \wedge A_j \wedge A_k\right) \\ & - \ldots + \ldots \\ & - \sum_{1 \leq i < j < \ldots l \leq n} \mathrm{P}\left(B \wedge A_i \wedge A_j \wedge \ldots \wedge A_l\right) \\ & + (-1)^{n-1} \mathrm{P}\left(B \wedge (\wedge_{i=1}^{n} A_i)\right) \Big] \Big/ \mathrm{P}\left((\vee_{i=1}^{n-1} A_i) \vee A_n\right) \end{aligned} \tag{C.1}$$

The proof by induction is as follows. First, the associative property can be applied to the denominator of Theorem 1 to obtain the following.

$$\mathrm{P}\left((\vee_{i=1}^{n-1} A_i) \vee A_n\right) = \mathrm{P}(\vee_{i=1}^{n} A_i) \tag{C.2}$$

Hence, the Inclusion-Exclusion Principle as shown in equation (C.1) can be applied to $\mathrm{P}(\vee_{i=1}^{n} A_i)$. Therefore, the problem statement is reduced to only the problem of the nominator of equation (C.1).

The base case is to consider a single set of $\mathrm{P}\left(B \wedge \left((\vee_{i=1}^{n-1} A_i) \vee A_n\right)\right)$, which is when $n = 1$. We can obtain:

$$\mathrm{P}\left(B \wedge \left((\vee_{i=1}^{0} A_i) \vee A_1\right)\right) = \mathrm{P}(B \wedge A_1) \tag{C.3}$$

Then, the Inclusion-Exclusion Principle is implied that

$$\mathrm{P}(B \wedge A_1) = \sum_{i=1}^{1} (B \wedge A_i) - \sum_{1 \le i < j \le n} \mathrm{P}\left(B \wedge A_i \wedge A_j\right) \tag{C.4}$$

, which is equal to itself because there is no such j that is greater than i and less than 1. Specifically for this problem, the base case should also be expanded to $n = 2$ as follow:

$$\begin{aligned} \mathrm{P}\left(B \wedge \left((\vee_{i=1}^{1} A_i) \vee A_2\right)\right) &= \mathrm{P}\left(B \wedge (A_1 \vee A_2)\right) \\ &= \sum_{i=1}^{2} \mathrm{P}\left(B \wedge A_i\right) - \sum_{1 \le i \le j \le 2} \mathrm{P}\left(B \wedge A_i \wedge A_j\right) \\ &= \mathrm{P}(B \wedge A_1) + \mathrm{P}(B \wedge A_2) - \mathrm{P}(B \wedge A_1 \wedge A_2) \end{aligned} \tag{C.5}$$

This is because the DCM-PA approach is developed to aggregate multiple domain-concepts, e.g. $\mathrm{P}(B \mid A_i \vee A_j)$. Further, the "Set Addition Principle" [31] states that:

$$A_1 \vee A_2 = (A_1 \setminus A_2) \vee (A_2 \setminus A_1) \vee (A_1 \wedge A_2) \tag{C.6}$$

, where all three components of the right hand side are disjoint and $(A_1 \setminus A_2)$ is a set whose members are in A_1, but not in A_2. Therefore, we can apply the Set Addition Principle to $\mathrm{P}(B \mid A_1 \vee A_2)$ in equation (C.5) to obtain:

$$
\begin{aligned}
\mathrm{P}\left(B \wedge (A_1 \vee A_2)\right) &= \mathrm{P}\left((B \wedge A_1) \vee (B \wedge A_2)\right) \\
&= \mathrm{P}\left((B \wedge A_1) \setminus (B \wedge A_2)\right) + \mathrm{P}\left((B \wedge A_2) \setminus (B \wedge A_1)\right) - \\
&\quad \mathrm{P}(B \wedge A_1 \wedge A_2) \\
&= \left[\mathrm{P}\left((B \wedge A_1) \setminus (B \wedge A_2)\right) + \mathrm{P}(B \wedge A_1 \wedge A_2)\right] + \\
&\quad \left[\mathrm{P}\left((B \wedge A_2) \setminus (B \wedge A_1)\right) + \mathrm{P}(B \wedge A_1 \wedge A_2)\right] - \\
&\quad \mathrm{P}(B \wedge A_1 \wedge A_2) \\
&= \mathrm{P}(B \wedge A_1) + \mathrm{P}(B \wedge A_2) - \mathrm{P}(B \wedge A_1 \wedge A_2)
\end{aligned}
\tag{C.7}
$$

This proofs that the Inclusion-Exclusion Principle holds for our base case of an aggregation of any two sets. The next step is to proof that Theorem 1 holds for any $n > 2$ using an induction hypothesis. Therefore, the hypothesis assumes that Theorem 1 holds for $1 \le i < m \le n$, which is:

$$
\begin{aligned}
\mathrm{P}\left(B \wedge \left((\vee_{i=1}^{n-1} A_i) \vee A_n\right)\right) &= \sum\nolimits_{i=1}^{n} \mathrm{P}\left(B \wedge A_i\right) \\
&\quad - \sum_{1 \le i < j \le n} \mathrm{P}\left(B \wedge A_i \wedge A_j\right) \\
&\quad + \sum_{1 \le i < j < k \le n} \mathrm{P}\left(B \wedge A_i \wedge A_j \wedge A_k\right) \\
&\quad - \ldots + \ldots \\
&\quad - \sum_{1 \le i < j < \ldots m \le n} \mathrm{P}\left(B \wedge A_i \wedge A_j \wedge \ldots \wedge A_n\right) \\
&\quad + (-1)^{n-1} \mathrm{P}\left(B \wedge (\wedge_{i=1}^{n} A_i)\right)
\end{aligned}
\tag{C.8}
$$

The above equation is from a problem of a two-set aggregation.

$$
B \wedge \left((\vee_{i=1}^{n-1} A_i) \vee A_n\right) = \left(B \wedge (\vee_{i=1}^{n-1} A_i)\right) \vee \left(B \wedge A_n\right) \tag{C.9}
$$

Therefore, we can rearrange the equation using the Set Addition Principle to obtain:

$$
\begin{aligned}
\mathrm{P}\left(B \wedge \left((\vee_{i=1}^{n-1} A_i) \vee A_n\right)\right) &= \mathrm{P}\left(B \wedge (\vee_{i=1}^{n-1} A_i)\right) + \mathrm{P}\left(B \wedge A_n\right) \\
&\quad - \mathrm{P}\left(\left(B \wedge (\vee_{i=1}^{n-1} A_i)\right) \wedge A_n\right)
\end{aligned}
\tag{C.10}
$$

At this step, let two groups of an arbitrary number of sets and their intersections be:

1. I_l is a collection of *k*-fold intersections of $(B \wedge A_1),(B \wedge A_2),...,(B \wedge A_{n-1})$.
2. $I_l^{'}$ is a collection of *k*-fold intersections of $(B \wedge A_1),(B \wedge A_2),...,(B \wedge A_n)$.

This implies that $(B \wedge A_n)$ is included every member of $I_l^{'}$, but not of I_l. Hence, these two collections do not duplicate.

From equation (C.10), we can obtain the following for *n* domain-concepts:

$$\begin{aligned} \mathrm{P}\left(B \wedge \left((\vee_{i=1}^{n-1} A_i) \vee A_n\right)\right) = \sum_{j=1}^{n-1}\left((-1)^{(j+1)} \sum_{S \in I_j} \mathrm{P}(S)\right) + \mathrm{P}\left(B \wedge A_n\right) \\ - \mathrm{P}\left(B \wedge (\vee_{i=1}^{n-1} A_i) \wedge A_n\right) \end{aligned} \quad \text{(C.11)}$$

Also, for the last component of the above equation, we can apply the set distributive property to obtain:

$$\mathrm{P}\left(B \wedge (\vee_{i=1}^{n-1} A_i) \wedge A_n\right) = \mathrm{P}\left(B \wedge \left(\vee_{i=1}^{n-1} (A_i \wedge A_n)\right)\right) \quad \text{(C.12)}$$

It is worth mentioning the following set operation will take care of some redundancies (if there are any):

$$(A_p \wedge A_r) \wedge (A_q \wedge A_r) = (A_p \wedge A_q \wedge A_r) \quad \text{(C.13)}$$

Therefore, the Inclusion-Exclusion Principle can also be applied to $\mathrm{P}\left(B \wedge \left(\vee_{i=1}^{n-1} (A_i \wedge A_n)\right)\right)$ because it is re-arranged to be a union problem of *n*-1 sets. Hence, we can obtain:

$$\mathrm{P}\left(B \wedge \left((\vee_{i=1}^{n-1} A_i) \vee A_n\right)\right) = \sum_{j=1}^{n-1}\left((\text{-}1)^{(j+1)} \sum_{S \in I_j} \mathrm{P}(S)\right) + \mathrm{P}\left(B \wedge A_n\right) - \sum_{j=1}^{n-1}\left((\text{-}1)^{(j+1)} \sum_{S \in I_j} \mathrm{P}(S \wedge A_n)\right) \quad \text{(C.14)}$$

, which resembles the union of two domain-concepts problem shown in equation (C.7). Further, the last component of (C.14) that has $\mathrm{P}(S \wedge A_n) \mid S \in I_j$ can be substituted by $\mathrm{P}(S) \mid S \in I'_{j+1}$ as follow:

$$\mathrm{P}\left(B \wedge \left((\vee_{i=1}^{n-1} A_i) \vee A_n\right)\right) = \sum_{j=1}^{n-1}\left((\text{-}1)^{(j+1)} \sum_{S \in I_j} \mathrm{P}(S)\right) + \mathrm{P}\left(B \wedge A_n\right) - \sum_{j=1}^{n-1}\left((\text{-}1)^{(j+1)} \sum_{S \in I'_{j+1}} \mathrm{P}(S)\right) \quad \text{(C.15)}$$

Moreover, the last component can also be re-written another way as its substitution shown below:

$$\mathrm{P}\left(B \wedge \left((\vee_{i=1}^{n-1} A_i) \vee A_n\right)\right) = \sum_{j=1}^{n-1}\left((\text{-}1)^{(j+1)} \sum_{S \in I_j} \mathrm{P}(S)\right) + \mathrm{P}\left(B \wedge A_n\right) - \sum_{j=2}^{n}\left((\text{-}1)^{(j)} \sum_{S \in I'_j} \mathrm{P}(S)\right) \quad \text{(C.16)}$$

The last component (+/-) sign has to be changed if the power of (-1) is changed from *j* to *j*+1.

$$\mathrm{P}\left(B \wedge \left((\vee_{i=1}^{n-1} A_i) \vee A_n\right)\right) = \sum_{j=1}^{n-1}\left((\text{-}1)^{(j+1)} \sum_{S \in I_j} \mathrm{P}(S)\right) + \mathrm{P}\left(B \wedge A_n\right) + \sum_{j=2}^{n}\left((\text{-}1)^{(j+1)} \sum_{S \in I'_j} \mathrm{P}(S)\right) \quad \text{(C.17)}$$

However, $(B \wedge A_n)$ can be included in the last component to obtain:

$$\begin{aligned} \mathrm{P}\left(B \wedge \left((\vee_{i=1}^{n-1} A_i) \vee A_n\right)\right) = \sum\nolimits_{j=1}^{n-1}\left((\text{-}1)^{(j+1)} \sum\nolimits_{S \in I_j} \mathrm{P}(S)\right) \\ + \sum\nolimits_{j=1}^{n}\left((\text{-}1)^{(j+1)} \sum\nolimits_{S \in I_j'} \mathrm{P}(S)\right) \end{aligned} \tag{C.18}$$

By combining two summations together, we can finally obtain:

$$\begin{aligned} \mathrm{P}\left(B \wedge \left((\vee_{i=1}^{n-1} A_i) \vee A_n\right)\right) = \sum\nolimits_{j=1}^{n}\left((\text{-}1)^{(j+1)}\left(\sum\nolimits_{S \in I_j} \mathrm{P}(S) + \sum\nolimits_{S \in I_j'} \mathrm{P}(S)\right)\right) \\ + (\text{-}1)^{(n-1)} \mathrm{P}\left(B \wedge (\wedge_{i=1}^{n} A_i)\right) \end{aligned} \tag{C.19}$$

Hence, equation (C.19) proofs the Inclusion-Exclusion Principle.

Q.E.D.

Theorem 2. The following statement is true for all $n \geq 1$.

$$\mathrm{P}\left(B \mid \left((\wedge_{i=1}^{n-1} A_i) \wedge A_n\right)\right) = \frac{\mathrm{P}\left(B \wedge \left((\wedge_{i=1}^{n-1} A_i) \wedge A_n\right)\right)}{\mathrm{P}\left((\wedge_{i=1}^{n-1} A_i) \wedge A_n\right)} \tag{C.20}$$

The proof by induction is as follows.

The associative property can be applied to Theorem 2 to obtain the following.

$$\begin{aligned} \mathrm{P}\left(B \mid \left((\wedge_{i=1}^{n-1} A_i) \wedge A_n\right)\right) &= \frac{\mathrm{P}\left(B \wedge \left(\wedge_{i=1}^{n-1} A_i \wedge A_n\right)\right)}{\mathrm{P}\left(\wedge_{i=1}^{n-1} A_i \wedge A_n\right)} \\ &= \frac{\mathrm{P}\left(B \wedge \left(\wedge_{i=1}^{n} A_i\right)\right)}{\mathrm{P}\left(\wedge_{i=1}^{n} A_i\right)} \end{aligned} \tag{C.21}$$

The base case of $\mathrm{P}\left(B \mid \left((\wedge_{i=1}^{n-1} A_i) \wedge A_n\right)\right)$ is when $n = 1$, which is:

$$\mathrm{P}(B \mid (\wedge_{i=1}^{0} A_i) \wedge A_1) = \frac{\mathrm{P}(B \wedge A_1)}{\mathrm{P}(A_1)} \tag{C.22}$$

Specifically for DCM-PA that aggregates two or more domain-concept partitions, the base case should also be expanded to $n = 2$, which is:

$$\begin{aligned} \mathrm{P}\left(B \mid \left((\wedge_{i=1}^{1} A_i) \wedge A_2\right)\right) &= \frac{\mathrm{P}\left(B \wedge (A_1 \wedge A_2)\right)}{\mathrm{P}(A_1 \wedge A_2)} \\ &= \frac{\mathrm{P}(B \wedge A_1 \wedge A_2)}{\mathrm{P}(A_1 \wedge A_2)} \end{aligned} \tag{C.23}$$

Therefore, the base case is true.

Assume the following holds for $1 \le i < m \le n$.

$$\mathrm{P}\left(B \mid \left((\wedge_{i=1}^{n-1} A_i) \wedge A_n\right)\right) = \frac{\mathrm{P}\left(B \wedge (\wedge_{i=1}^{n-1} A_i) \wedge A_n\right)}{\mathrm{P}\left((\wedge_{i=1}^{n-1} A_i) \wedge A_n\right)} \tag{C.24}$$

Further, based on the set theorem and the associative property of an intersection operation, the following also holds for $1 \le i < m \le n$.

$$\mathrm{P}\left((\wedge_{i=1}^{n-1} A_i) \wedge A_n\right) = \mathrm{P}(\wedge_{i=1}^{n} A_i) \tag{C.25}$$

Given the above is true, prove the following.

$$\mathrm{P}\left(B \mid \left((\wedge_{i=1}^{n} A_i) \wedge A_{n+1}\right)\right) = \frac{\mathrm{P}\left(B \wedge \left((\wedge_{i=1}^{n} A_i) \wedge A_{n+1}\right)\right)}{\mathrm{P}\left((\wedge_{i=1}^{n} A_i) \wedge A_{n+1}\right)} \tag{C.26}$$

By set association, we can obtain:

$$\mathrm{P}\left((\wedge_{i=1}^{n} A_i) \wedge A_{n+1}\right) = \mathrm{P}(\wedge_{i=1}^{n+1} A_i) \tag{C.27}$$

Finally, we can obtain:

$$\mathrm{P}\left(B \mid \left(\left(\wedge_{i=1}^{n} A_i\right) \wedge A_{n+1}\right)\right) = \frac{\mathrm{P}\left(B \wedge \left(\wedge_{i=1}^{n+1} A_i\right)\right)}{\mathrm{P}\left(\wedge_{i=1}^{n+1} A_i\right)} \tag{C.28}$$

Q.E.D.

BIBLIOGRAPHY

[1] M. H. Dunham, *Data Mining: Introductory and Advanced Topics*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall/Pearson Education, 2003.

[2] A. Ceglar and J. F. Roddick, "Association Mining," *ACM Computing Surveys (CSUR),* vol. 38, pp. 1-42, 2006.

[3] Oracle Corporation, "Oracle Data Mining Concepts 10g Release 1 (10.1)," Oracle Database Document Library, 2003.

[4] D.-I. Lin and Z. M. Kedem, "Pincer-Search: An Efficient Algorithm for Discvoering the Maximum Frequent Set," *IEEE Transactions on Knowledge and Data Engineering (TKDE),* vol. 14, pp. 553-66, 2002.

[5] R. J. Bayardo Jr and R. Agrawal, "Mining the Most Interesting Rules," in *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, 1999, pp. 145-54.

[6] J. F. Roddick and S. Rice, "What's Interesting about Cricket? On Thresholds and Anticipation in Discovered Rules," *ACM SIGKDD Explorations Newsletter,* vol. 3, pp. 1-5, 2001.

[7] A. A. Freitas, "Are We Really Discovering "Interesting" Knowledge from Data?," in *Expert Update (the BCS-SGAI Magazine)*. vol. 9, 2006, pp. 41-7.

[8] D. T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining*. Hoboken, NJ: John Wiley & Sons, Inc., 2005.

[9] J. Hsu, "Critical and Future Trends in Data Mining: A Review of Key Data Mining Technologies/Applications," in *Data Mining Opportunities and Challenges*, J. Wang, Ed. Hershey, PA: Idea Group Publishing (an imprint of Idea Group Inc.), 2003, pp. 437-52.

[10] M. Kantardzic, *Data Mining. Concepts, Models, Methods, and Algorithms*. Piscataway, NJ: IEEE Press, 2003.

[11] A. A. Freitas and S. H. Lavington, "Speeding Up Knowledge Discovery in Large Relational Databases by Means of a New Discretization Algorithm," in *Advances in Databases (Proceedings of the 14th British National Conference on Databases - BNCOD-14)*, R. Morrison and J. Kennedy, Eds. Edinburgh, UK: Springer-Verlag, 1996, pp. 124-33.

[12] R. Agrawal, T. Imielinski, and A. Swami, "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, 1993, pp. 207-16.

[13] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," in *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, Santiago, Chile, 1994, pp. 487-99.

[14] Y. Aumann and Y. Lindell, "A Statistical Theory for Quantitative Association Rules," in *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '99)*, San Diego, CA, USA, 1999, pp. 261-70.

[15] S. Brin, R. Motwani, and C. Silverstein, "Beyond Market Baskets: Generalizing Association Rules to Correlations," in *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, Tucson, AZ, 1997, pp. 324-38.

[16] T. Y. Lin, X. Hu, and E. Louie, "A Fast Association Rule Algorithm Based on Bitmap and Granular Computing," in *The 12th IEEE Conference on Fuzzy Systems 2003 (FUZZ '03)*, Saint Louis, MO, USA, 2003, pp. 678-83.

[17] H. Toivonen, "Sampling Large Databases for Association Rules," in *Proceedings of the 22th International Conference on Very Large Data Bases*, 1996, pp. 134-45.

[18] H. Mannila, H. Toivonen, and A. I. Verkamo, "Efficient Algorithms for Discovering Association Rules," in *Proceedings AAAI'94 Workshop Knowledge Discovery in Databases (KDD'94)*, Seattle, WA, 1995, pp. 181-92.

[19] B. Lent, A. Swami, and J. Widom, "Clustering Association Rules," in *Proceedings of the 13th International Conference on Data Engineering (ICDE'97)*, 1997, pp. 220-31.

[20] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns Without Candidate Generation: A Frequent-Pattern Tree Approach," *Data Mining and Knowledge Discovery,* vol. 8, pp. 53-87, 2004.

[21] C. Lucchese, S. Orlando, and R. Perego, "Fast and Memory Efficient Mining of Frequent Closed Itemsets," *IEEE Transactions on Knowledge and Data Engineering (TKDE),* vol. 18, pp. 21-36, 2006.

[22] L. Bing, W. Hsu, and Y. Ma, "Pruning and Summarizing the Discovered Associations," in *Proceeding of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '99)*, San Diego, CA, USA, 1999, pp. 125-34.

[23] M. Dai and Y.-L. Huang, "Organizing the Discovered Association Rules Based on General-Specific (GS) Hierarchical Patterns," in *Proceedings of the 4th International Conference on Machine Learning and Cybernetics*, Guanzhou, China, 2005, pp. 2206-11.

[24] A. Berrado and G. C. Runger, "Using Metarules to Organize and Group Discovered Association Rules," *Data Mining and Knowledge Discovery,* vol. 14, pp. 409-31, 2007.

[25] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo, "Finding Interesting Rules from Large Sets of Discovered Association Rules," in *Proceedings of the 3rd International Conference on Information and Knowledge Management*, Gaithersburg, MD, 1994, pp. 401-7.

[26] B. Liu, W. Hsu, and Y. Ma, "Mining Association Rules with Multiple Minimum Supports," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '99)*, San Diego, CA, USA, 1999, pp. 337-41.

[27] Y.-L. Cheung and A. W.-C. Fu, "Mining Frequent Itemsets Without Support Threshold: With and Without Item Constraints," *IEEE Transactions on Knowledge and Data Engineering (TKDE),* vol. 16, pp. 1052-69, 2004.

[28] R. Srikant and R. Agrawal, "Mining Quantitative Association Rules in Large Relational Tables," in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (ACM SIGMOD '96)*, Montreal, Canada, 1996, pp. 1-12.

[29] W. K. Mahamaneerat, T. Kobayashi, and J. M. Green, "Domain-Concept Mining on the 2005 Nationwide Inpatient Sample Data," in *American Medical Informatics Association (AMIA) 2007 Annual Symposium Data Mining Competition* Chicago, IL, 2007.

[30] A. Silberschatz and A. Tuzhilin, "What Makes Patterns Interesting in Knowledge Discovery Systems," *IEEE Transactions on Knowledge and Data Engineering (TKDE),* vol. 8, pp. 97-4, 1996.

[31] J. S. Milton and J. C. Arnold, *Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences*, 3rd ed. New York: Irwin/McGraw-Hill, 1995.

[32] B. Rosner, *Fundamentals of Biostatistics*, 5th ed. Boston, MA: Duxbury Press, 2006.

[33] The Centers for Disease Control and Prevention (CDC), "BRFSS - 2006 Survey Data," Atlanta, GA: U.S. Department of Health and Human Services, 2006.

[34] R. Ramakrishnan and J. Gehrke, *Database Management Systems*, 3rd ed. Boston, MA: McGraw Hill, 2003.

[35] Healthcare Cost and Utilization Project (HCUP), "HCUP-US Overview," in *Nationwide Inpatient Sample*: Agency for Healthcare Research and Quality (AHRQ), 2008.

[36] The Centers for Disease Control and Prevention (CDC), "Behavioral Risk Factor Surveillance System Survey Data," Atlanta, GA: U.S. Department of Health and Human Services, 1990-2006.

[37] J. D. Wren, "Extending the Mutual Information Measure to Rank Inferred Literature Relationships," *BMC Bioinformatics,* vol. 5, 2004.

[38] D. J. Hand, "Statistics and Data Mining: Intersecting Disciplines," *ACM Explorations Newsletter - Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD'99),* vol. 1, pp. 16-19, 1999.

[39] Y. Xu, S.-X. Zhou, and J.-H. Gong, "Mining Association Rules with New Measure Criteria," in *Proceedings of the 4th International Conference on Machine Learning and Cybernetics*, Guangzhou, China, 2005, pp. 2257-60.

[40] W.-G. Teng, M.-S. Chen, and P. S. Yu, "A Regression-Based Temporal Pattern Mining Scheme for Data Streams," in *Proceedings of the 29th ACM VLDB International Conference on Very Large Data Bases (VLDB' 03).* 2003, pp. 93-104.

[41] Y. Ke, J. Cheng, and W. Ng., "Mining Quantitative Correlated Patterns Using and Information-Theoretic Approach," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*, Philadelphia, PA, 2006, pp. 227-36.

[42] E. Achtert, C. Bohm, H.-P. Kriegel, P. Kroger, and A. Zimek, "Deriving Quantitative Models for Correlation Clusters," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*, Philadelphia, Pennsylvania, USA, 2006, pp. 4-13.

[43] T. Calders, B. Goethais, and S. Jaroszewicz, "Mining Rank-Correlated Sets of Numerical Attributes," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*, Philadelphia, PA, 2006, pp. 96-105.

[44] Y. Leung, J.-H. Ma, and W.-X. Zhang., "A New Method for Mining Regression Classes in Large Data Sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, pp. 6-21, 2001.

[45] R. J. Bayardo Jr. and R. Agrawal, "Mining the Most Interesting Rules," in *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '99)*, San Diego, CA, USA, 1999, pp. 145-54.

[46] G. Grahne and J. Zhu, "Fast Algorithms for Frequent Itemset Mining Using FP-Trees," *IEEE Transactions on Knowledge and Data Engineering (TKDE),* vol. 17, pp. 1347-62, 2005.

[47] J. Pei and J. Han, "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach," *IEEE Transactions on Knowledge and Data Engineering (TKDE),* vol. 16, pp. 1424-40, 2004.

[48] G. Li and H. J. Hamilton, "Searching for Pattern Rules," in *The 6th IEEE International Conference on Data Mining (ICDM'06)*, Hong Kong, China, 2006, pp. 933-37.

[49] M. J. Zaki and C.-J. Hsiao, "Efficient Algorithms for Mining Closed Itemset and Their Lattice Structure," *IEEE Transactions on Knowledge and Data Engineering (TKDE),* vol. 17, pp. 462-78, 2005.

[50] M. J. Zaki, "Generating Non-Redundant Association Rules," in *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Boston, MA, 2000, pp. 34-43.

[51] H.-C. Chang, C.-C. Hsu, and E. Chen, "Mining Closed Frequent Itemsets for Incremental and Diminished Database with Lexicographic Tree Traversal," in *Proceedings of Networks, Parallel and Distributed Processing, and Applications (NPDPA 2002)*, Tsukuba, Japan, 2002.

[52] V. Pudi and J. R. Haritsa, "Generalized Closed Itemsets for Association Rule Mining," in *Proceedings of the International Conference on Data Engineering*, Bangalore, India, 2003, pp. 714-6.

[53] A. Gionis, H. Mannila, T. Mielikainen, and P. Tsaparas, "Accessing Data Mining Results via Swap Randomization," in *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining (KDD'06)*, Philadelphia, PA, USA, 2006, pp. 167-76.

[54] J. Han, Q. Yang, and E. Kim, "Plan Mining by Divide-and-Conquer," in *Proceedings 1999 SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'99)* Philadelphia, PA, 1999, pp. 1-6.

[55] C.-H. Lee, C.-R. Lin, and M.-S. Chen, "Sliding-Window Filtering: An Efficient Algorithm for Incremental Mining," in *Proceedings of the ACM 10th International Conference on Information and Knowledge Management (CIKM-01)*, 2001, pp. 263-70.

[56] A. Schuster and R. Wolff, "Communication-Efficient Distributed Mining of Association Rules," *Data Mining and Knowledge Discovery,* vol. 8, pp. 171-96, 2004.

[57] S. Brin, R. Rastogi, and K. Shim, "Mining Optimized Gain Rules for Numeric Attributes," *IEEE Transactions on Knowledge and Data Engineering (TKDE),* vol. 15, pp. 324-38, 2003.

[58] J.-M. Wei, W.-G. Yi, M.-Y. Wang, and S.-Q. Wang, "Data Feature Oriented Data Partition and Weighted Data Mining," in *Proceedings of the International Conference on Information Acquisition*, 2004, pp. 287-91.

[59] B. Chen, P. Haas, and P. Scheuermann, "A New Two-Phase Sampling Based Algorithm for Discovering Association Rules," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, 2002, pp. 462-8.

[60] B. Thuraisingham, L. Khan, C. Clifton, J. Maurer, and M. Ceruti, "Dependable Real-Time Data Mining," in *Proceedings of the 8th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05)*, 2005, pp. 1-8.

[61] V. S. Tseng, C.-J. Chu, and T. Liang, "Efficient Mining of Temporal High Utility Itemsets from Data Streams," in *ACM KDD Workshop on Utility-Based Data Mining (UDBM'06)*, Philadelphia, PA, USA, 2006, pp. 18-27.

[62] W.-J. Lee, J.-Y. Jiang, and S.-J. Lee, "An Efficient Algorithm to Discover Calendar-Based Temporal Association Rules," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2004, pp. 3122-7.

[63] R. Nock, P.-A. Laur, and J.-E. Symphor, "Statistical Borders for Incremental Mining," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, 2006, pp. 212-5.

[64] V. Ng, S. Chan, D. Lau, and C. M. Ying, "Incremental Mining for Temporal Association Rules for Crime Pattern Discoveries," in *Proceedings of the 8th Australasian Database Conference (ADC2007)*, 2007, pp. 123-32.

[65] W. G. Aref, M. G. Elfeky, and A. K. Elmagarmid, "Incremental, Online, and Merge Mining of Partial Periodic Patterns in Time-Series Databases," *IEEE Transactions on Knowledge and Data Engineering (TKDE),* vol. 16, pp. 332-42, 2004.

[66] E. R. Omiecinski, "Alternative Interest Measures for Mining Associations in Databases," *IEEE Transactions on Knowledge and Data Engineering (TKDE),* vol. 15, pp. 57-69, 2003.

[67] M. Zaki, S. Parthasarathy, W. Li, and M. Ogihara, "Evaluation of Sampling for Data Mining of Association Rules," in *Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE'97)*, Birmingham, UK, 1997, pp. 42-50.

[68] J. Dong and M. Han, "BitTableFI: An Efficient Mining Frequent Itemsets Algorithm," *Knowledge-Based System,* vol. 20, pp. 329-35, 2007.

[69] B. K. Sy and A. K. Gupta, *Information-Statistical Data Mining: Warehouse Integration with Examples of Oracle Basics*. Norwell, MA, USA: Kluwer Academic Publishers, 2004.

[70] C. E. Shannon, *The Mathematical Theory of Communication*. Urbana, Illinois: Urbana, University of Illinois Press, 1964.

[71] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York: ACM Press, 1999.

[72] J. Peat and B. Barton, *Medical Statistics: A Guide to Data Analysis and Critical Appraisal*, 1st ed. MA: Blackwell Publishing, 2005.

[73] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Boston, MA: McGraw-Hill Book Company, 2001.

[74] K. Gouda and M. J. Zaki, "GenMax: An Efficient Algorithm for Mining Maximal Frequent Itemsets," *Data Mining and Knowledge Discovery,* vol. 11, pp. 1-20, 2005.

[75] D.-L. Yang, C.-T. Pan, and Y.-C. Chung, "An Efficient Hash-Based Method for Discovering the Maximal Frequent Set " in *Proceedings of the 25th Annual International Computer Software and Applications Conference (COMPSAC'01)* Chicago, IL, 2001, pp. 511-6.

[76] E. J. M. Lauria and G. K. Tayi, "Bayesian Data Mining and Knowledge Discovery," in *Data Mining Opportunities and Challenges*, R. R. Johnson, Ed. Harshey, PA: Idea Group Publishing (an imprint of Idea Group Inc.), 2003, pp. 260-277.

[77] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. San Diego, CA: Academic Press, Inc., 1990.

[78] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA: Morgan Kaufmann Publishers Inc., 1993.

[79] W.-J. Lee and S.-J. Lee, "A General Mining Method for Incremental Updation in Large Databases," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2003, pp. 1423-8.

[80] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*. Boca Raton, Florida: Chapman & Hall/CRC, 1995.

[81] B. Wilson, "The Harmonic Mean." vol. 2006 Sydney, Australia: School of Computer Science and Engineering (CSE), The University of New South Wales (UNSW). Online: http://www.cse.unsw.edu.au/~teachadmin/info/harmonic3.html. Accessed: June 2, 2006, 2007.

[82] J. Eng, "Sample Size Estimation: How Many Individuals Should Be Studied?," *Radiology,* vol. 227, pp. 309-313, 2003.

[83] W. G. Hopkins, "New View of Statistics: Effect Magnitudes," in *A New View of Statistics*: Internet Society for Sport Science. Online: http://www.sportsci.org/ resource/stats/effectmag.html. Accessed: April 10, 2007, 1997.

[84] F. Faul, E. Erdfelder, A.-G. Lang, and A. Buchner, "G*Power 3: A flexible Statistical Power Analysis Program for the Social, Behavioral, and Biomedical Sciences," *Behavioral Research Methods,* vol. 39, pp. 175-191, 2007.

[85] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, revised ed. New York: Academic Press, 1977, 1977.

[86] W. G. Hopkins, "Measures of Reliability in Sports Medicine and Science," *Sports Medicine,* vol. 30, pp. 1-15, 2000.

[87] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Hillsdale, NJ: Lawrence Earlbaum Associates, 1988.

[88] L. Comtet, *Advance Combinatorics: the Art of Finite and Infinite Expansions*. Boston, MA: D. Reidel Publisher Company, 1974.

[89] B. A. Davey and H. A. Priestley, *Introduction to Lattices and Order*, 2nd ed. Cambridge, U.K.: Cambridge University Press, 2002.

[90] W. V. Quine, *Set Theory and Its Logic*. Cambridge, MA: Belknap Press of Harvard University Press, 1969.

[91] Wolfram Mathematica, "Union -- from Wolfram MathWorld," Wolfram Research, Inc. Online: http://mathworld.wolfram.com/Union.html. Accessed: March 12, 2008, 2008.

[92] C.-R. Shyu, M. Klaric, G. Scott, and W. K. Mahamaneerat, "Knowledge Discovery by Mining Association Rules and Temporal-Spatial Information from Large-Scale Geospatial Image Databases," in *Proceedings of the International Geoscience and Remote Sensing Symposium*, Denver, CO, 2006.

[93] Google Inc., "Google Earth," Mountain View, CA, 2008.

[94] W. K. Mahamaneerat, L. A. Snow, and C.-R. Shyu, "Mining Associations from the CDC's Behavioral Risk Factor Surveillance System Database to Assist Policy Making," in *Enhancing Healthcare Education, Research & Practice Symposium, A Special Track of the 2007 International Conference on ICT in Teaching and Learning* Hong Kong, China, 2007.

[95] W. K. Mahamaneerat, C.-R. Shyu, B. R. Stewart, and J. M. Armer, "Post-op Swelling and Lymphoedema Following Breast Cancer Treatment: A Baseline-Comparison BMI-Adjusted Approach," *Journal of Lymphoedema,* 2008 (accepted).

[96] W. K. Mahamaneerat, C.-R. Shyu, S.-C. Ho, and C. A. Chang, "Domain-Concept Association Rules Mining for Large Scale and Complex Cellular Manufacturing Tasks," *Journal of Manufacturing Technology Management,* vol. 18, pp. 787-806, 2007.

[97] I. Mullins, M. Siadaty, J. Lyman, K. Scully, C. Garrett, W. M. WG, R. Muller, B. R. B, C. Apte, S. Weiss, I. Rigoutsos, D. Platt, S. Cohen, and W. Knaus, "Data Mining and Clinical Data Repositories: Insights from a 667,000 Patient Data Set," *Computers in Biology and Medicine,* vol. 36, pp. 1351-77, 2006.

[98] N. Lavrač, M. Bohanec, A. Pur, B. Cestnik, M. Debeljak, and A. Kobler, "Data Mining and Visualization for Decision Support and Modeling of Public Health-Care Resources," *Journal of Biomedical Informatics,* vol. 40, pp. 438-447, 2007.

[99] American Cancer Society (ACS), "Cancer Facts & Figures 2006," Atlanta, GA 2007.

[100] Office for National Statistics, "Cancer Statistics Registrations: Registrations of Cancer Diagnosed in 2004, England," National Statistics, London, UK 2007.

[101] American Cancer Society (ACS), "Lymphedema Understanding and Managing Lymphedema after Cancer Treatment," Atlanta, GA 2006.

[102] J. Ferlay, F. Bray, P. Pisani, and D. Parkin, "GLOBOCAN 2002: Cancer Incidence, Mortality and Prevalence Worldwide," in *IARC Cancer Base No. 5, Version 2.0* Lyon, France, 2004.

[103] J. Disa and J. Petrek, "Rehabilitation After Treatment for Cancer of the Breast," in *Cancer Principles and Practice of Oncology*, 6th ed, V. T. DeVita, Jr., S. Hellman, and S. A. Rosenberg, Eds. PA: Lippincott, Williams & Wilkins, 2001, pp. 1717-26.

[104] J. M. Armer and B. R. Stewart, "A Comparison of Four Diagnostic Criteria for Lymphedema in a Post-Breast Cancer Population," *Lymphatic Research and Biology,* vol. 3, pp. 208-17, 2005.

[105] J. M. Armer, "The Problem of Post-Breast Cancer Lymphedema: Impact and Measurement Issues," *Cancer Investigations,* vol. 1, pp. 76-83, 2005.

[106] J. Petrek and M. Heelan, "Incidence of Breast Carcinoma-Related Lymphedema," *Cancer Epidemiology, Biomarkers & Prevention,* vol. 83, pp. 2776-81, 1998.

[107] S. Passik, and M. McDonald, "Psychosocial Aspects of Upper Extremity Lymphedema in Women Treated for Breast Carcinoma," *American Cancer Society Lymphedema Workshop: Supplement to Cancer,* pp. 2817-20, 1998.

[108] S. G. Rockson, "Diagnosis and Management of Lymphedema," *Cancer Supplement,* vol. 83, pp. 2882-2885, 1998.

[109] A. G. Meek, "Breast Radiotherapy and Lymphedema," *Cancer Supplement,* vol. 83, pp. 2788-97, 1998.

[110] C. F. Petlund, "Volumetry of Limbs," in *Lymph Stasis: Pathophysiology, Diagnosis and Treatment*, W. L. Olszewski, Ed. Boca Raton: CRC Press, 1991, pp. 443-52.

[111] J. M. Armer and M. Fu, "Age Differences in Post-Breast Cancer Lymphedema Signs and Symptoms," *Cancer Nursing,* vol. 28, pp. 200-7, 2005.

[112] J. Armer, M. R. Fu, J. M. Wainstock, E. Zagar, and L. K. Jacobs, "Lymphedema Following Breast Cancer Treatment, Including Sentinel Lymph Node Biopsy," *Lymphology,* vol. 37, pp. 73-91, 2004.

[113] J. J. Coen, A. G. Taghian, L. A. Kachnic, S. I. Assaad, and S. N. Powell, "Risk of Lymphedema After Regional Nodal Irradiation with Breast Conservation Therapy," *International Journal of Radiation Oncology, Biology, Physics,* vol. 55, pp. 1209-15, 2003.

[114] M. Deutsch and J. Flickinger, "Arm Edema After Lumpectomy and Breast Irradiation," *American Journal of Clinical Oncology,* vol. 26, pp. 229-31, 2003.

[115] B. M. Geller, P. M. Vacek, P. O'Brien, and R. H. Secker-Walker, "Factors Associated with Arm Swelling After Breast Cancer Surgery," *Journal of Women's Health,* vol. 12, pp. 921-30, 2003.

[116] C. Ozaslan and B. Kuru, "Lymphedema After Treatment of Breast Cancer," *The American Journal of Surgery,* vol. 187, pp. 69-72, 2004.

[117] A. C. Voogd, J. M. Ververs, A. J. Vingerhoets, R. M. Roumen, J. W. Coebergh, and M. A. Crommelin, "Lymphoedema and Reduced Shoulder Function as Indicators of Quality of Life After Axillary Lymph Node Dissection for Invasive Breast Cancer," *British Journal of Surgery,* vol. 90, pp. 76-81, 2003.

[118] J. M. Armer, W. K. Mahamaneerat, T. Kobayashi, O. Nukaew, B. R. Stewart, and C.-R. Shyu, "Occurrence of Lymphedema Following Breast Cancer Treatment using BMI-Adjusted Volume Change," in *The 21st International Congress of Lymphology 2007* Shanghai, China, 2007 (in press).

[119] H. Feigelson, C. Jonas, L. Teras, M. Thun, and E. Calle, "Weight Gain, Body Mass Index, Hormone Replacement Therapy, and Postmenopausal Breast Cancer in a Large Prospective Study," *Cancer Epidemiology, Biomarkers & Prevention,* vol. 13, pp. 220-4, 2004.

[120] R. Chlebowski, E. Aiello, and A. McTiernan, "Weight Loss in Breast Cancer Patient Management," *Journal of Clinical Oncology,* vol. 20, pp. 1128-43, 2002.

[121] K. O. Johansson, K.; Ingvar, M.; Albertson, M.; Ekdahl, C., "Factors Associated with the Development of Arm Lymphedema Following Breast Cancer Treatment: A Match Pair Case-Control Study," *Lymphology,* vol. 35, pp. 59-71, 2002.

[122] M. Whiteman, S. Hillis, K. Curtis, J. McDonald, P. Wingo, and P. Marchbanks, "Body Mass and Mortality After Breast Cancer Diagnosis," *Cancer Epidemiology, Biomarkers & Prevention,* vol. 14, pp. 2009-14, 2005.

[123] J. A. Petrek, R. T. Senie, M. Peters, and P. P. Rosen, "Lymphedema in a Cohort of Breast Carcinoma Survivors 20 Years After Diagnosis," *Cancer,* vol. 92, pp. 1368-77, 2001.

[124] A. Soran, G. D'Angelo, M. Begovic, F. Ardic, A. Harlak, H. S. Wieand, V. G. Vogel, and R. R. Johnson, "Breast Cancer-Related Lymphedema--What are the Significant Predictors and How They Affect the Severity of Lymphedema?," *Breast Journal,* vol. 12, pp. 536-43, 2006.

[125] C. W. Callaway, "Circumferences," in *Anthropometric Standardization Reference Manual*, T. G. Lohman, A. F. Roche, and R. Martorell, Eds. Illinois: Human Kinetics Books, 1988, pp. 39-54.

[126] Centers for Disease Control and Prevention (CDC), "Body Mass Index: BMI for Adults," Atlanta, GA: U.S. Department of Health and Human Services, 2008.

[127] P. Brink and M. Wood, *Advanced Design in Nursing Research*, 2nd ed. Thousand Oaks, CA: Sage Publications, 1998.

[128] P. Pudil, J. Novovicova, and J. Kittler, "Floating Search Methods in Feature Selection," *Pattern Recognition Letters,* vol. 15, pp. 1119-25, 1994.

[129] A. Jain and D. Zonger, "Feature Selection: Evaluation, Application, and Small Sample Performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, pp. 153-8, 1997.

[130] K. Fukunaga, "Feature Extraction and Linear Mapping for Classification: Feature Subset Selection," in *Introduction to Statistical Pattern Recognition*, 2nd ed San Diego, CA: Academic Press, Inc, 1990, p. 491.

[131] R. G. Askin and C. R. Standridge, *Modeling and Analysis of Manufacturing Systems*. New York: John Wiley & Sons, Inc, 1993.

[132] C. Dimopoulos and N. Mort, "A Hierarchical Clustering Methodology Based on Genetic Programming for the Solution of Simple Cell-Formation Problems," *International Journal of Production Research,* vol. 39, pp. 1-19, 2001.

[133] G. Harhalakis, G. Ioannou, I. Minis, and R. Nagi, "Manufacturing Cell Formation Under Random Product Demand," *International Journal of Production Research,* vol. 32, pp. 47-64, 1994.

[134] D. Cao and M. Chen, "A Robust Cell Formation Approach for Varing Product Demands," *International Journal of Production Research,* vol. 43, pp. 1587-1605, 2005.

[135] J. Balakrishnan and C. H. Cheng, "Dynamic Cellular Manufacturing Under Multiperiod Planning Horizons," *Journal of Manufacturing Technology Management,* vol. 16, pp. 516-30, 2005.

[136] S. Sofianopoulou, "Manufacturing Cells Efficiency Evaluation Using Data Envelopment Analysis," *Journal of Manufacturing Technology Management,* vol. 17, pp. 224-38, 2006.

[137] J. L. Burbidge, "Production Flow Analysis," *Production Engineering,* vol. 42, pp. 742-52, 1963.

[138] N. R. des Santos and L. O. de Araújo Jr., "Computaional System for Group Technology - PFA Case Study," *Integrated Manufacturing Systems,* vol. 14, pp. 138-52, 2003.

[139] J. R. King, "Machine-Component Grouping in Production Flow Analysis: An Approach Using a Rank Order Clustering Algorithm," *International Journal of Production Research,* vol. 18, pp. 117-33, 1980.

[140] W. T. McCormick, R. J. Schweitzer, and T. W. White, "Problem Decomposition an Data Reorganization by Clustering Techniques," *Operations Research,* vol. 20, pp. 993-1009, 1972.

[141] M. P. Chandrasekharan and R. Rajagopalan, "MODROC: An Extension of Rank Order Clustering for Group Technology," *International Journal of Production Research,* vol. 24, pp. 1211-33, 1986.

[142] J. McAuley, "Machine Grouping for Efficient Production," *Production Engineering,* vol. 51, pp. 53-7, 1972.

[143] H. M. Selim, R. M. S. A. Aal, and A. I. Mahdi, "Formation of Machine Group and Part Families: A Modified SLC Method and Comparative Study," *Integrated Manufacturing Systems,* vol. 14, pp. 123-37, 2003.

[144] T. Gupta, "Clustering Algorithms for the Design of a Cellular Manufacturing System - An Analysis of Their Performance," *Computer and Industrial Engineering,* vol. 20, pp. 461-68, 1991.

[145] M. P. Chandrasekharan and R. Rajagopalan, "An Ideal Seed Non-Hierarchical Clustering Algorithm for Cellular Manufacturing," *International Journal of Production Research,* vol. 24, pp. 451-64, 1986.

[146] M. P. Chandrasekharan and R. Rajagopalan, "ZODIAC- An Algorithm for Concurrent Formation of Part Families and Machine-Cells," *International Journal of Production Research,* vol. 25, pp. 835-50, 1987.

[147] G. Srinivasan and T. T. Narendran, "GRAFICS- A Non-Hierarchical Clustering Algorithm for Group Technology," *International Journal of Production Research,* vol. 29, pp. 463-78, 1991.

[148] J. Miltenburg and W. Zhang, "A Comparative Evaluation of Nine Well-Known Algorithms for Solving the Cell Formation Problem in Group Technology," *Journal of Operations Management,* vol. 10, pp. 44-72, 1991.

[149] C.-H. Chu and J. C. Hayya, "A Fuzzy-Clustering Approach to Manufacturing Cell Formation," *Journal of Production Research,* vol. 29, pp. 1475-87, 1991.

[150] J. A. Joines, C. T. Culbreth, and R. E. King, "Manufacturing Cell Design: An Integer Programming Model Employing Genetic Algorithms," *IIE Transactions,* vol. 28, pp. 69-85, 1996.

[151] C. H. Cheng, Y. P. Gupta, W. H. Lee, and K. F. Wong, "A TSP-Based Heuristic for Forming Machine Groups and Part Families," *International Journal of Production Research,* vol. 36, pp. 1325-37, 1998.

[152] S. Zolfaghari and M. Liang, "Comprehensive Machine Cell/Part Family Formation Using Genetic Algorithm," *Journal of Manufacturing Technology Management,* vol. 15, pp. 433-44, 2004.

[153] S. Zolfaghari and E. V. L. Roa, "Cellular Manufacturing Versus a Hybrid System: A Comparative Study," *Journal of Manufacturing Technology Management,* vol. 17, pp. 942-61, 2006.

[154] M. C. Chen, "Configuration of Cellular Manufacturing Systems Using Association Rule Induction," *International Journal of Production Research,* vol. 41, pp. 381-95, 2003.

[155] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Greedy Algorithms," in *Introduction to Algorithms* Saint Louis, MO: McGraw-Hill, 1998, pp. 329-55.

[156] C. S. Kumar and M. P. Chandrasekharan, "Grouping Efficacy: A Quantitative Criterion for Goodness of Block Diagonal Forms of Binary Matrices in Group Technology," *International Journal of Production Research,* vol. 28, pp. 603-12, 1990.

[157] F. F. Boctor, "A Linear Formulation of the Machine-Part Cell Formation Problem," *International Journal of Production Research,* vol. 29, pp. 343-56, 1991.

[158] W. J. Boe and C. H. Cheng, "A Close Neighbour Algorithm for Designing Cellular Manufacturing Systems," *International Journal of Production Research,* vol. 29, pp. 2097-116, 1991.

[159] J. L. Burbidge, *The Introduction of Group Technology*. New York, NY: Wiley, 1975.

[160] A. S. Carrie, "Numerical Taxonomy Applied to Group Technology and Plant Layout," *International Journal of Production Research,* vol. 11, pp. 339-416, 1973.

[161] M. P. Chandrasekharan and R. Rajagopalan, "GROUPABILITY: An Analysis of the Properties of Binary Data Matrices for Group Technology," *International Journal of Production Research,* vol. 27, pp. 1035-52, 1989.

[162] H. Seifoddini, "Single Linkage vs Average Linkage Clustering in Machine Cells Formation Application," *Computer and Industrial Engineering,* vol. 16, pp. 419-26, 1989.

[163] Siemens AG and ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, "KDD Cup 2008," Las Vegas, NV, 2008.

[164] R. Kaye, "Open Induction, Tennenbaum Phenomena, and Compexity Theory," in *Arithmetic, Proof Theory, and Computational Complexity*, P. Clote and J. Krajícek, Eds. Oxford, England: Clarendon Press, 1992, pp. 222-237.

[165] M. Slone and K. Ferguson, "Proof of Principle of Inclusion-Exclusion," PlanetMath. Online: http://planetmath.org/?op=getobj&from=objects&id=2804. Accessed: April 20, 2008, 2005.

VITA

Wannapa *Kay* Mahamaneerat (วรรณภา มหามณีรัตน์) was born in Bangkok, Thailand in 1974. She holds a BA degree in Economics with a minor in Applied Computer Science (1995) and a MS degree in Applied Statistics with a specialty in Information Systems Management (1997) from Thailand, and a MS degree in Computer Science (2001) from the University of Missouri (MU). Kay's areas of expertise include information retrieval, database, and data mining in public health and biomedical informatics. Her research work includes, but is not limited to, data mining on the Behavioral Risk Factor Surveillance System for the Centers for Disease Control and Prevention, Nationwide Inpatient Sample research, breast cancer survivors with lymphedema data sets, gene pathway extraction from texts, and microarray gene expression patterns for cancer classification.

Kay's research projects have been recognized as a winner of the "AMIA 2007 Data Mining Competition" and she has earned first place in the "2007 Computer Science Graduate Student Council Annual Poster Competition". Also in March 2008, she was named one of the "Outstanding Graduate Students" by the MU College of Engineering. In June 2008, she was a recipient of the MU Graduate School John Bies International Travel Scholarship for ABD Doctoral Student. She is first and co-author of numerous publications from the lymphedema research project, as well as other data mining publications.