Faculty of Engineering and Information Technology
University of Technology, Sydney

Mining Actionable Combined Patterns Satisfied both Utility and Frequency Criteria

A thesis submitted in partial fulfillment of
the requirements for the degree of
Master of Analytics by Research

by

Jingyu Shao

June 2016

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

Acknowledgments

Foremost, I would like to express my sincere gratitude to my supervisors Prof. Longbing Cao for the continuous support of my master degree study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor during my study.

I also would like to appreciate my co-supervisor Doctor Guandong Xu and Wei Liu for providing me with continuous support throughout my PhD study and research. Without their professional guidance and persistent help, this thesis would not have been possible.

I thank my fellow labmates in Advanced Analystics Institute: Junfu Yin, Xiangfu Meng, Xing Wang, Shoujin Wang, Xuhui Fan, CC Chen, Jia Xu and Liang Hu for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last two years.

Last but not the least, I would like to thank my family: my father and my mother, for their unconditional support, both financially and emotionally throughout the whole master studying.

Jingyu
November 2015 @ UTS

Contents

List of Figures

List of Tables

List of Publications

Papers Published

- **Jingyu Shao**, Junfu Yin, Wei Liu, Longbing Cao (2015), Mining Actionable Combined Patterns of High Utility and Frequency. *in* 'Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics (**DSAA15**)', full paper accepted.

- **Jingyu Shao**, Junfu Yin, Wei Liu, Longbing Cao (2015), Actionable Combined High Utility Itemset Mining. *in* 'Proceedings of the 29th Association for the Advancement of Artificial Intelligence (**AAAI15**)', poster accepted.

- Xiangfu Meng, Longbing Cao, **Jingyu Shao** (2014), Semantic Approximate Keyword Query Based on Keyword and Query Coupling Relationship Analysis. *in* 'Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (**CIKM14**)' , pp. 529-538.

- Xiangfu Meng, Longbing Cao, **Jingyu Shao** (2014), Finding Top-k Semantically Related Terms in Relational Keyword Search. *in* 'Proceedings of the 2014 IEEE International Conference on Data Science and Advanced Analytics (**DSAA14**)', pp. 505-511.

Papers to be Submitted/Under Review

- **Jingyu Shao**, Junfu Yin, Longbing Cao (2015), Mining Strong Associated Patterns among High Utility Itemsets. "To be submitted".

Abstract

In the last two decades, researchers have proposed numerous approaches and techniques for extracting frequent patterns. Until recent ten years, researchers have not realized the disadvantages of mining frequent patterns in several cases. One paradoxical case is that in a digital store, laptops with quite low frequency earn much higher profit than memory disks which have a high frequency. To tackle such issues, the relative importance of each item has been introduced into frequent pattern mining, and the concept "high utility itemsets mining" has been proposed. The criteria for discovering high utility patterns is a user-specified minimum utility threshold, instead of a minimum support threshold, to extract itemsets with high utilities. Even though the introduction of utility can solve some business issues better than the frequency-based measurements, the resultant patterns are still not actionable in tackling business concerns.

Accordingly, actionable knowledge discovery is proposed to identify informative and decision-making-friendly knowledge that satisfies both technical and business criteria to narrow the large gap between technically identified results and real-world user needs. Actionable pattern mining has proved to be essential for handling those impact-targeted activities and business problems, such as behavior analysis, fraud detection and government-customer debt. In addition, it has an outstanding performance especially in imbalanced datasets. For example, one of the key business concerns in the activity pattern analysis is to find out which particular activity directly triggers or is closely associated with the occurrence of a target impact.

During recent years, actionable knowledge discovery has demonstrated its value in solving business and industrial concerns, where the analysis of pattern relationship plays a foundational role, and combined pattern mining is the basic approach to generate such kind of knowledge. One approach is to develop the utility framework that is more suitable for addressing business consideration than the frequency framework, while none of existing work has been reported on discovering actionable knowledge from utility databases. Hence, it is essential to build an applicable approach for mining actionable combined patterns from utility datasets. However, there are challenges for achieving so. 1) The downward closure property does not hold in utility-based mining approaches, which means that most of the existing algorithms for frequency-based mining cannot be applied. 2) Furthermore, compared to high utility mining methods, actionable combined knowledge discovery faces the critical combinational complexity as well as the complicated structure caused by the dependence between items.

In order to address these research limitations and challenges, this thesis proposes an actionable combined knowledge discovery framework for mining actionable combined patterns that satisfy both utility and frequency requirements. The thesis is organized as follows.

Chapter 2 briefly reviews the related works on the frequent pattern mining framework, high utility itemset mining framework, and the actionable knowledge discovery approach. Chapter 3 incorporates the utility concept into combined pattern mining, and actionable patterns with high utility growth and strong associations are defined and discovered. An efficient algorithm called CUARM (Combined Utility-Association Rule Mining) is presented for actionable high utility pattern mining. A basic tree structure for mining utility growth patterns is proposed, and a measure considering both utility growth and co-occurrence rate is proposed to finalize the discovery of such combined patterns. Chapter 4 discusses how to discover those high utility patterns with highly associated relationship between one item and another. Such patterns have a significant feature, that is the utility increases with the

length of such pattern increasing. That is to say, the utilities of the derivative itemsets are always higher than those of underlying itemsets. Also, a hybrid algorithm for mining both highly dependent and utility growth patterns is proposed to obtain those highly dependent actionable patterns.

All of the algorithms are examined in both synthetic and real datasets, and their performance is compared with baselines for mining frequent patterns and high utility patterns. The results show that our proposed actionable combined patterns are more informative for business decision-support.

Chapter 1

Introduction

1.1 Background

Frequent patterns can be seen everywhere in our daily life. The concept of "frequency" has wide application. For example, when you visit a restaurant that you have never been to before, you may first check all the recommended dishes on the menu. The restaurant highlights the "recommended" dishes based on the frequency of all the dishes ordered. Wordy examples are abandoned in frequent mining because quite a huge number of extensions and applications have been studied since it was first proposed by Agrawal (Agrawal, Srikant et al. 1994) in 1994.

Utility is an important concept in economics, because it represents the satisfaction experienced by the consumer in relation to one product. In the cross-domain of information technology and business, utility is a measurement in terms of representing the interestingness of items (itemsets) such as products, stocks, customers and so on. For example, concerning businessmen, the utility can be the profit of the product, the cost of running a business or the implementation of a strategy. Concerning customers, the utility measures the value or impact of customer behaviors, e.g., the profit rate of a shopping behavior, or signifies the satisfaction degree of customer interactions with shopping mall or the ranks to a popular film. While fre-

quency mining is a traditional topic in information technology, utility mining is a cross-domain of IT and economics. Like frequency mining, the utility dataset contains members (also called items, elements) and original details such as the transaction ID and transaction time. Unlike frequency mining, a utility dataset contains more specific information about members such as unit profits and the quantities of each product.

A variety of applications take utility into account. Typical examples include the profit of products in supermarkets and retail stores, the satisfaction feedbacks of different restaurants, the popularity of hot showing movies and even frauds.

We illustrate two cases in detail below to demonstrate the application of frequency and utility.

The first case is the on-line shopping website. Nowadays, on-line shopping websites such as Amazon, eBay, Taobao and Groupon are increasingly popular and well-known. Customers tend to buy goods online rather than purchase stuff at a physical store due to the discounted price, convenience, various choices and other advantages. Retailers also prefer to open a virtual store because of low overheads and easy management. However, these websites have to deal with the numbers in terms of access everyday. One of the backstage tasks is to record the customer behaviour such as the clicks and scrolls to a web log database, as shown in Table 1.1. The behaviour or action of a visitor is described in each row i.e. the user ID, from where they have logged on the web, the exact page URL and the action. For example, U_1 may notice a recommendation from his friend ABC about scuba diving when he is reading on Twitter. After he clicks it, the browser automatically opens a new website displaying the scuba diving. The following two rows record his behaviour in relation to purchasing this event. All these actions are captured by Groupon's servers behind the web pages, and stored in their web log database. As to user U_3, she experienced a considerable hesitation period in realtion to making a decision on yoga training. She first saw the same recommendation about scuba diving from the same friend as user U_1.

On another occasion, she found an interesting yoga course but she hesitated as to whether to purchase it or not. Another time, after she found an additional recommendation about this yoga course on Facebook, she decided to attend the event.

Table 1.1: The Web Access Log

User-ID	Referring URL	Page URL	Action
U_1	www.twitter.com/user-ID=ABC	www.groupon.com/view_scuba-diving	View
U_1	...	www.groupon.com/checkout_scuba-diving	Checkout
U_1	...	www.groupon.com/purchase_scuba-diving	Purchase
U_2	www.twitter.com/user-ID=ABC	www.groupon.com/view_scuba-diving	View
U_3	www.twitter.com/user-ID = ABC	www.groupon.com/view_scuba-diving	View
U_3	...	www.groupon.com/view_yoga	View
U_3	...	www.groupon.com/checkout_yoga	Checkout
U_3	www.facebook.com/user-ID = xyz	www.groupon.com/view_yoga	View
U_3	...	www.groupon.com/checkout_yoga	Checkout
U_3	...	www.groupon.com/purchase_yoga	Purchase
U_4	www.facebook.com/user-ID = xyz	www.groupon.com/view_horse-riding	View
U_4	...	www.groupon.com/view_yoga	View
U_4	...	www.groupon.com/view_scuba-diving	View

Figure 1.1: The shopping basket

Website data analysts are keen to know which items are the most popular ones and which items should be recommended to a group of related friends. The behaviours of user U_4 could be helpful to illustrate such objectives. When user U_4 opened the web page about horse riding referred to on Facebook, he noticed that one of his friends had purchased a yoga course. After he finished reading the horse riding web page, he turned to the yoga webpage to peruse the details. The Groupon's server also recommended a popular event: scuba diving, as all the other users U_1, U_2 and U_3 have visited this event.

The second case is a transaction record including different customer shopping baskets, as shown in Table 1.2. This table is sorted from a retail store's transaction database which contains customers' transaction records and detailed information. The first column is the Transaction IDs (TID), which are uniquely assigned to the related transactions. The second column demonstrates the exact time for transactions. The products one customer purchased are listed in the third column, followed by the quantity of each purchased product and the unit profit of each product in the last two column. Each row can be regarded as a customer's purchased shopping basket as shown in Fig. 1.1.

The objective of a retailer or a supermarket manager is to increase the

Table 1.2: A Transaction Record Table

TID	Transaction Time	Products	Quantities	Unit Profit
T_1	23-09-2015 10:00:43	33	1	$ 16.80
T_2	23-09-2015 10:03:22	16, 47, 55	2, 3, 1	$ 3.30, $ 1.20, $ 1.80
T_3	23-09-2015 10:08:55	28, 29	1,3	$ 1.40, $ 2.20
T_4	23-09-2015 10:22:51	21	2	$ 1.50
T_5	23-09-2015 10:32:03	16	2	$ 3.30
...	...	...	...	...
T_{1393}	30-09-2015 22:47:53	16, 55	1, 4	$ 3.30, $ 1.80

total turnover and profit of each product. To achieve this goal, the shopping habits of a variety of customers should be studied to present a well-behaved selling and promotion strategy. The data collected from the transaction record database should be helpful for managers to learn and discover such knowledge and find out the valuable patterns based on customers' behaviours. The promotion strategies may then be designed in terms of the discovered patterns. Consequently, the revenue is improved.

1.2 Actionable Knowledge Discovery

All the algorithms and approaches above are proposed for academic purposes, and focus on the discovery of patterns that satisfy the expected standard and significance such as the frequency and utility. However, as Cao et al. discusses in their book (Cao, Philip, Zhang & Zhao 2010), such discovered patterns often cannot support real users' needs in relation to taking actions on various domains for business and industrial purposes. This is because of the absence of informative knowledge. The main reasons why the results from academic approaches are not actionable and cannot be applied to business

are as follows:

- Business interestingness is rarely considered in current pattern mining. Even though there are often many patterns captured via efficient algorithms, these patterns are always useless to business people because they have been generated in a general manner, and it is difficult to capture and satisfy particular business expectations.

- Very limited preliminary work has been conducted on developing subjective and business oriented interest measures, and little of this aims at a standard and general measurement.

- There is an issue in developing business interest metrics. As results generated in a general manner cannot satisfy business interestingness, a more practical way is to cater for specific expectations in terms of a real mining approach, and to generate business interestingness metrics. However, this is a difficult approach because on one hand, general approaches ignore the subjective and objective interest measures. On the other hand, business people are not the ones that major in building mathematical models. Regardless of the interesting metrics, it is hard to figure out how to interpret the findings and what straightforward actions can be taken to support business decision-making and operations.

As mentioned above, there is a large gap between academic deliverables and business objectives. For example, Cao's research (Cao, Zhao & Zhang 2008) talks about mining impacted-targeted activity patterns from imbalanced data and its business objectives are to discover debt-oriented activity patterns. However, these patterns are at a lower frequency than academic patterns. By contrast, those non-debt-oriented patterns which display that both the degree of support and confidence are high have proved to be meaningless. To this end, it is necessary to develop effective mining methods for business and industrial purposes to deal with not only the special expectations, but also the imbalanced data. Therefore, Cao et al. have

proposed AKD (Actionable Knowledge Discovery) (Cao, Zhao, Figueiredo, Ou & Luo 2007, Cao et al. 2008, Cao, Zhang, Zhao, Luo & Zhang 2011) and DDDM (Domain Driven Data Mining) (Cao et al. 2010).

1.3 Combined Pattern Mining

Actionable knowledge discovery is aimed at involving and satisfying business objectives. However, enterprise data mining applications inevitably involve complex data sources. In addition, there is an increasing need for mining complex knowledge on the accumulation of ubiquitous enterprise data for informative decision-making. To achieve actionable knowledge, a combined mining approach would be helpful.

However, mining more informative and decision-making oriented patterns composed of multiple features and the characteristics of information from multiple sources and datasets is far from a trivial task. It actually presents many critical challenges.

- Patterns which are discovered by traditional algorithms only involve homogeneous features from a single source of data. Such single source patterns contain limited information and are not useful for business decision-making.
- In addition, generally speaking, it is not only costly but also space consuming to merge multiple, heterogeneous and large numbers of sources together for pattern analysis and knowledge discovery. Sometimes, this might be even impossible because of the memory limitations and so on.
- Single measurement is often not that powerful and efficient to generate actionable patterns in mining multiple heterogeneous datasets. As a result, these patterns are not sufficient to solve real-world comprehensive issues in business and industrial domains.

To overcome the existing challenges in mining single measurement-based patterns from single sources, Cao and his group have proposed the concepts of

combined association rules, *combined rule pairs* and *combined rule clusters* to cater for the comprehensive aspects reflected through variety measurements from multiple datasets (Zhao, Zhang, Cao, Zhang & Bohlscheid 2008, Zhang, Zhao, Cao & Zhang 2008, Cao et al. 2011, Cao 2012). A combined association rule is composed of multiple featured items or atomic patterns which are generated from different datasets via different measurements. In addition, combined pattern pairs and combined pattern clusters are built from combined association rules.

The delivery of combined patterns represents an in-depth and more comprehensive indicator for decision-making actions and makes the patterns more informative and actionable than patterns composed of only single aspects or identified by single measurement-based results. This is because combined patterns consist of multiple components and a pair or cluster (Cao et al. 2011) of atomic patterns, identified in individual sources or based on individual measurements.

Actionable knowledge discovery, with the combined pattern mining method, is a closed optimisation problem solving process, it incorporates issue and concept definitions and a framework and model design to real-world problem solutions through delivery decision-making oriented patterns. These are highly related to business and industrial processes and systems. Following the presentation of this idea, we present the limitation and challenges of the current actionable combined pattern mining in the next section.

1.4 Limitations and Challenges

Although the actionable knowledge discovery approach as well as combined pattern mining algorithms successfully extract patterns from application datasets in order to solve real business and industrial problems to some degree, their only interest measurement and concepts such as local support, lift etc. are based on the frequency of patterns. In other words, any frequent pattern is treated as the same significant, one. However, in practice,

most frequent patterns (no matter if they are actionable or not) may only be applied in limited circumstances, where different attributes or features are treated as the same importance. There are also circumstances where truly interesting patterns cannot be detected via frequency measurements. For example, in fraud detection, the stock market and even in retail stores, decisions should not only be made on the basis of frequency. In addition, some actionable patterns are not as informative as others, and they could obstruct the really useful decision-making. In retail business, for example, selling a laptop usually leads to a higher profit than selling a memory card, while the frequency of selling laptops is much lower than that of memory cards. Even though at times customers who purchase a laptop might also like to buy a memory card, this can cause a weak association between them. As to online banking fraud detection, a large amount of money transfers to an unauthorised account may seldom appear, it can nonetheless have a strong business impact.

General approaches cannot solve such problems in a satisfactory way. In a related domain, the relative importance of each item is not considered in frequent pattern mining(FPM). To tackle this issue, weighted association rules mining has been proposed by (Cai, Fu, Cheng & Kwong 1998, Wang, Yang & Yu 2000, Tao, Murtagh & Farid 2003, Yun & Leggett 2005*a*, Yun 2007*a*, Yun 2008*a*). In this framework, different weighted values are associated with each item, for example, the unit profit of each product and the unit cost of each transportation. Those less frequent itemsets could also be discovered owing to this framework. However, this approach still ignores the gap between academic and real business because it is not simply the unit profit but also the purchased quantity which should be considered in the business domain and the requirements of retailers who are interested in discovering itemsets with high sale profits cannot be satisfied by this approach. Utility mining emerges as an important topic in the data mining field. Mining high utility itemsets from databases refers to mining itemsets with a high level of interest, high importance or high profitability. The utility of an item in

a transaction database is comprised of two aspects: external utility, which is the interestingness of distinct items; internal utility, which is the interestingness of an item in specific transactions. The utility of an itemset in the whole dataset is the product of its internal utility and its external utility. Of course, a high utility itemset is the utility of this itemset no less than a used defined threshold called the minimum utility threshold.

While utility is introduced into the frequent pattern mining framework to discover patterns of high utility by considering the quality as well as the quantity of any product, it makes high utility itemsets mining (Yao, Hamilton & Butz 2004) a possible method by selecting interesting patterns based on the minimum utility threshold instead of the minimum support (Liu, Liao & Choudhary 2005, Li, Huang, Chen, Liu & Lee 2008, Ahmed, Tanbeer, Jeong & Lee 2009, Wu, Fournier-Viger, Yu & Tseng 2011, Liu, Wang & Fung 2012, Liu & Qu 2012, Wu, Shie, Tseng & Yu 2012). However, treating the utility as the only measurement still makes the results not actionable at all. For example, in some conditions, a high utility transaction appears accidentally, such as a purebred pet dog is purchased with a collar, food, soft mat and so on. As a purebred dog can be treated as a very high utility item, any combination of items including this purebred dog can be high utility itemsets in a pet store. However, such a transaction happens only once in several months, which cannot help the retailer to make a good decision as to whether they should stock more purebred pets or rare pets. This case shows the big gap between academic research and real business purposes, which drives researchers to find new ways to tackle these real world problems, especially for business purposes and industrial needs.

More details of the utility-based mining algorithms are shown in Section 2.2, and Section 2.3 in Chapter 2.

1.5 Research Issues

Based on the aforementioned current research limitations, we present the following research issues:

1.5.1 Mining Actionable Combined Patterns in High Utility Itemsets

For the process of mining actionable patterns with both high utility increment and high frequency, we summarise the issues below:

- The *downward closure property*(Agrawal et al. 1994) cannot be applied in the utility-based framework. The utility of one itemset is neither monotonic nor anti-monotonic while the length of the itemset changes, which means the utility of the itemset might be either higher or lower compared with its superset or subset. In addition, for a tree-structured algorithm, it is hard to assert which branch has the highest utility until all the branches are calculated. Thus, fast algorithms such as in (Han, Pei & Yin 2000) cannot be applied to mining HUI.

- In high utility itemsets mining, a large number of candidates would be generated if a lower threshold is given. By contrast, if the threshold is set too high, only the absolutely high utility itemsets can be discovered, and it is then hard to identify the most profitable itemsets for a given item. Subsequently, when utility is considered as the only metric to select patterns, high utility pattern mining may result in findings that are not typical and do not consider the couplings between items.

- While the combination of utility mining with frequent pattern mining is promising, the question is how to combine the utility framework with association rule mining. Association rule mining cares about the co-occurrence relationship between items based on the supports of items, while it is not clear how to measure the itemset associations for high

utility items. An item's utility depends not only on the quantity of an item in a transaction, but also its item utility.

1.5.2 Mining Highly Dependent Patterns in High Utility Itemsets

For the process of mining actionable patterns with both a high utility increment and a strong association, we summarise the issues below:

- Utility is not the only criterion for mining actionable patterns from a utility database. As the utility of a supper set might be higher or lower than its original itemset, and there is no evidence to prove that the change of utility follows a specific rule, taking the utility as the only measurement might deliver confusing and misleading results. For instance, a transaction which happened in a pet store included a purebred dogs, dog food, a collar and toys proved to be a high utility itemset because the utility of a purebred dog is beyond comparison. In this case, there is a large utility gap between the underlying itemset (dog toys) and its derivative itemset (purebred dog and toys), which makes this underlying itemset not actionable at all.

- The relationship between items should be taken into consideration for delivering decision-making results for business people. Various algorithms and approaches are proposed for mining the sorts of relationships between items. An alternative means to discover actionable patterns is the combined pattern mining method. However, this approach is only utilised in frequency-based frameworks, and no existing research is established for mining actionable patterns from a utility database.

1.6 Research Contributions

1.6.1 Mining Combined Patterns of Both High Utility and High Frequency

We have such contributions for the discovery of those combined patterns of both high utility and high frequency.

- A novel pattern structure, called *Actionable Combined Utility-Association Rule (CUAR)*, is proposed. The proposed pattern structure enables the generation of patterns that are of high utility and strongly associated. This occurs through the consideration of relationships between items, which provide users with actionable knowledge.
- A new level of interestingness for selecting patterns, called *Associated-Utility Growth (AUG)*, which integrates the relationship (association) and utility is proposed. To our knowledge, it is the first method for selecting patterns that have high utility without losing the representativeness (namely strong association).
- Intensive experiments on both synthetic and real datasets are conducted to evaluate the proposed methods.

1.6.2 Mining Strongly Associated Patterns in High Utility Incremental Itemsets

We also summarise the contributions when we are mining the strong association and high utility increment patterns.

- We propose a combined framework for discovering actionable knowledge from a utility database. Based on a series of novel definitions such as *utility growth* which have never been used in previous work, we theoretically prove that the proposed representation is effective and the patterns discovered are actually actionable and useful for decision-making.

- An efficient algorithm called MHUSAP(Mining High Utility and Strong Association Patterns) is proposed for mining actionable patterns with both high utility increment and high dependence. We systematically analyse the relationship between underlying itemsets and derivative itemsets based on both the utility and frequency criteria.

- Two effective strategies are applied to enhance the performance of MHUSAP. Based on the framework, we propose a global pruning strategy to generate the utility increment patterns from underlying itemsets. An equation is also proposed as a local approach for selecting patterns with the highest weighted values, which is a hybrid measurement of both utility growth and dependence calculation.

1.7 Thesis Structure

The thesis is structured as follows:

Chapter 2 provides the literature review of the definitions of frequency and utility, fast algorithms for frequent pattern mining and high utility itemset mining. Normal, top-k, closed and sequence approaches are represented with their advantages.

Chapter 3 incorporates an actionable combined framework into utility pattern mining, and a new approach for mining both high utility and high frequency patterns is defined. The efficient algorithm called CUARM (Combined Utility Association Rules Mining) is proposed to achieve this goal. In CUARM, we introduce a refined UG-tree for the first strategy to reduce the candidate space. Only utility growth patterns can be constructed in this tree. Furthermore, we present an equation to measure the interestingness of both frequency and utility in order to capture the actionable combined patterns satisfying both utility and frequency criteria. Substantial experiments on both synthetic and real datasets show that CUARM efficiently identifies actionable combined patterns from large scale data.

Chapter 4 builds another framework which combines of association dis-

covery and utility increment to mine associated utility-incremental patterns. An efficient algorithm named MHUSAP (Mining High Utility and Strong Association Patterns) is proposed to discover such patterns. In MHUSAP, we introduce an improved UG-tree as a one-step strategy to select those utility-incremental patterns. Furthermore, we present an indicator to measure the value of combined coefficient which can be regarded as a weight of confidence and utility increment.

Chapter 5 concludes the thesis and outlines the scope for future work.

Figure 1.2 shows the research profile of this thesis.

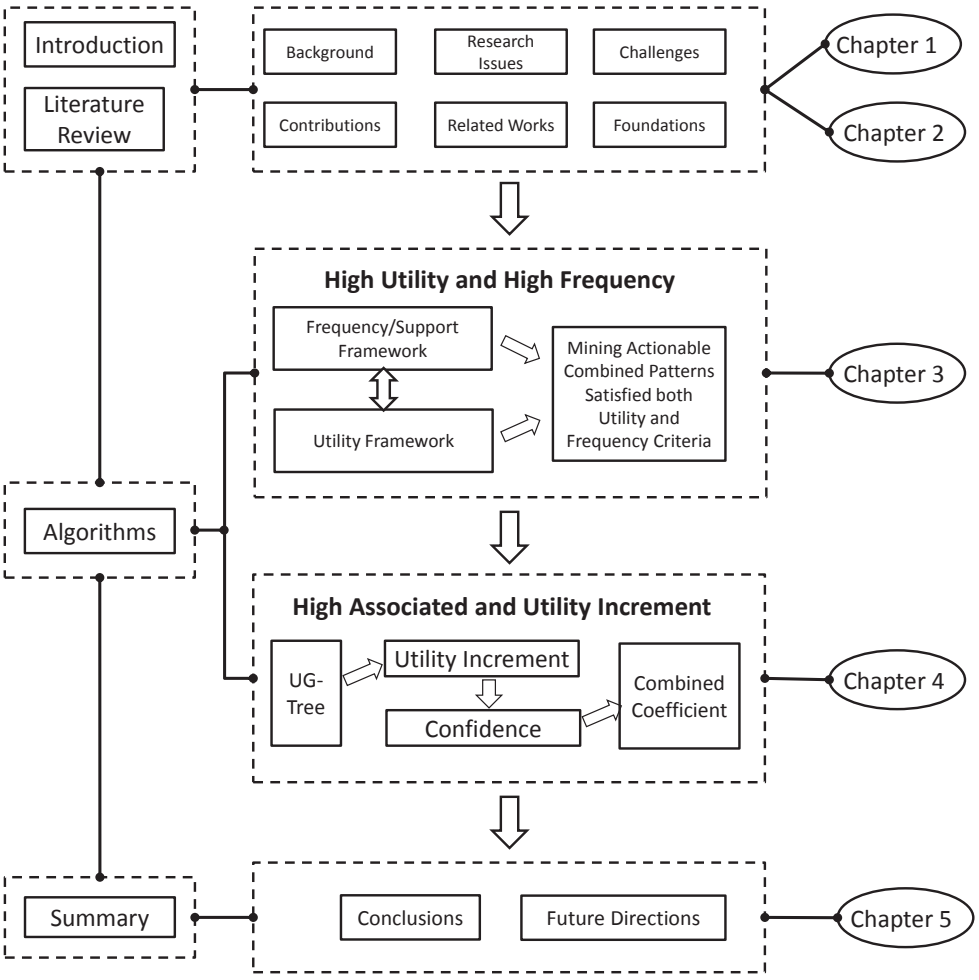


Figure 1.2: The profile of work in this thesis

Chapter 2

Literature Review and Foundation

Several kinds of related work are reviewed in this chapter. Firstly, in section 1, the widely recognised Frequent Itemset Mining (FIM) in the data mining area is demonstrated, with its concept, foundational property, fast algorithms and applications. Then the concept of High Utility Itemset Mining (HUIM) is introduced, which is an extension and is more interesting in the business domain. In the third section, a new framework called actionable combined mining is presented, which helps businessmen to make actionable decisions easier and more reasonable via a novel pattern discovering method.

2.1 Frequent Pattern Mining

Mining frequent itemsets (Agrawal et al. 1994) is a primary research topic and has been fully extended into diversified directions (Han, Cheng, Xin & Yan 2007) for a score of years since it was first introduced in 1994 by Rakesh Agrawal et al. for market basket analysis in order to discover the association rules. It is intended to identify strong rules discovered in databases using different measures of interestingness. Based on the concept of strong rules, Rakesh Agrawal et al. (Agrawal et al. 1994) introduced association rules

for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets. After this, a variety of algorithms were proposed to tackle the efficiency and accuracy scalable issue. Among them, *FP-Growth* proposed by Han et al. (Han et al. 2000) is one of the most efficient algorithms in FIM. In addition, abundant related work has been studied on this research topic which varies from the basic efficiency and accuracy improvement algorithms for frequent itemset mining in given datasets to the countless research extensions such as frequent sequential pattern mining (Yin, Zheng & Cao 2012), pattern mining from imbalanced data (Cao et al. 2008), structured pattern mining, associative classification, and frequent pattern-based clustering. Many more frontiers with their extended applications are established and these are hard to describe in one survey. Frequent pattern mining research is one of the basic issues in the data mining domain and is the foundation of data analysis, which , even though it has been studied for decades, is still having a deep impact on methodologies and extensions in the long run.

Let $D = \{T_1, T_2, ..., T_m\}$ be a transaction database, and $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ be a set of finite and discriminative items. Each transaction $T_c \in D$ $(1 < c < m)$ is a subset of $\mathcal{I}$ with a distinct identifier called *TID*. An itemset X containing k items is called a k-itemset X, which must belong to one of the transactions. In addition, X is called a frequent itemset if it appears no less than $\theta \times |D|$ times in a transaction database, where θ is set by the user, called *minimum support threshold* and denoted as *min_sup*. $|D|$ is the number of transactions in D.

Definition 2.1 *The **frequency** of an itemset X counts the times it appears in all transactions and is denoted as $SC(X)$, which is also called the **support count**. The **support** of X is $SC(X)/|D|$, and is denoted as $supp(X)$.*

The relative support range is in [0, 1]. The introduction of support greatly weakens the effect of support count demonstration when a large transaction database is used and it is useful for support comparisons.

Table 2.1: Transaction Database from Retail Store

TID	Transaction
T_1	bread, milk
T_2	bread, milk, butter, cheese
T_3	bread, milk, cheese
T_4	bread, butter, cheese
T_5	milk, butter

For example, we assume the itemset $\mathcal{I}$ including all products sold in a retail store is $\mathcal{I} = \{bread, milk, butter, cheese\}$ and present this in Table 2.1. "TID" is a unique transaction identification in the database and it describes the exact items in each transaction.

From Table 2.1, for example, we learn that the support of bread is $Supp(bread) = \frac{4}{5} = 80\%$, the support of both bread and milk is $Supp(bread, milk) = \frac{3}{5} = 60\%$. Here, $|D|$ is equal to 5 because there are 5 transactions in the database.

Agrawal and Srikant proposed the *Downward Closure Property (DCP)*, which is also known as *Apriori Property* in 1994 (Agrawal et al. 1994).

Property 2.1 *An itemset X is a frequent itemset if and only if all its sub-itemsets are frequent.*

This property also reveals that if X is a frequent itemset, then, any sub-itemset of X must be frequent. On the other hand, the super-itemset of a frequent itemset should not be a frequent itemset. For example, assuming itemset {a, b, c} is a frequent itemset, all its sub-itemsets such as {a, b}, {c} should also be frequent; if itemset {d} is an infrequent itemset, its super-

itemsets e.g. {a, d}, {b, d, e} are also infrequent itemsets.

This property is also the essence of the Apriori algorithm. In the first step, all frequent 1-itemsets (itemsets with only one item) can be discovered by scanning the transaction database, then using such itemsets to generate candidate 2-itemsets, and scanning the database again to pick out the frequent 2-itemsets. The process is repeated until all the frequent itemsets are mined, which means no more candidates can be generated.

Besides this, extensive algorithms are proposed as improvements and extensions based on Apriori. Park et al. proposed a hashing technique based on effective algorithm (Park, Chen & Yu 1995*a*). One algorithm based on partitioning was proposed by Savasere et al., see also 1995 (Savasere, Omiecinski & Navathe 1995). After that, for association rules mining, there is incremental mining (Cheung, Han, Ng & Wong 1996) along with parallel and distributed mining (Park, Chen & Yu 1995*b*, Agrawal & Shafer 1996, Cheung, Han, Ng, Fu & Fu 1996, Zaki, Parthasarathy, Ogihara & Li 1997) etc.

The algorithms above are called extensions and improvements of Apriori because they are all candidate-generating algorithms, which can suffer time consuming bacause of two major nontrivial costs: (1) the generation of large numbers of candidate itemsets, and (2) the repeated scanning of the original database as well as checking the support base of each candidate. Due to the shortages of Apriori, Han et al. proposed a novel structure of the pattern mining algorithm called FP-Growth (Han et al. 2000). Based on a divide-and-conquer strategy, the algorithm FP-Growth has become widely recognised because of its efficiency, which may be attributed to its execution without the need for candidate generation. A Trie data structure called Frequent-Pattern tree(FP-Tree) is the foundation of this algorithm. More details of this algorithm will be demonstrated later.

2.1.1 Association Rules Mining

Rakesh Agrawal et al. were commonly recognised as the first to propose the association rule as well as the FIM. One association rule is composed of

two parts with a linking symbol "→", each part is an itemset denoted as X and Y, where $X \cap Y = \emptyset$. An association rule can be understood by the form $\{X \rightarrow Y\}$, where $X \subseteq \mathcal{I}$, $Y \subseteq \mathcal{I}$ are called antecedent and consequent (Agrawal et al. 1993).

Generally speaking, one association rule is an itemset that all items contained obey a given rule. Numerous measurements are proposed for catching the interestingness of the association rules. Among them is the best known minimum support-confidence threshold constraint, which means that the pattern that we discover for the association rule should match two thresholds, being minimum support and minimum confidence.

- Support is a percentage, a ratio of $SC(X \rightarrow Y)/|D|$, equals to its support count divide the total transaction number. For example, the support of association rule $\{X \rightarrow Y\}$ is denoted as $supp(X \rightarrow Y)$.

- Reveal one kind of rule relation between antecedent and consequent, and is defined as follows.

$$conf(X \rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)} \tag{2.1}$$

  $Sup(X \rightarrow Y)$ is the support of itemset {X, Y}, which indicates both antecedent and consequent. The value of confidence shows how the consequent associated with antecedent. In addition, the range of confidence is also in [0, 1].

Still, we take Table 2.1 as an example. We can get $Supp(bread) = 80\%$, $Supp(bread, milk) = 60\%$, thus the confidence of $\{bread \rightarrow milk\}$ can be calculated as $Conf(bread \rightarrow milk) = \frac{Supp(bread,milk)}{Supp(bread)} = \frac{0.6}{0.8} = 75\%$. On the other hand, the confidence of $\{milk \rightarrow bread\}$ is $Conf(milk \rightarrow bread) = \frac{Supp(bread,milk)}{Supp(milk)} = \frac{0.6}{0.8} = 75\%$, which means customers who purchased milk would like to get some bread as well. In addition, when it refers to $\{bread, milk \rightarrow butter\}$, as the support of {bread, milk, butter} $Supp(bread, milk, butter) = 20\%$, the confidence of it is $Conf(bread, milk \rightarrow butter) = 33.3\%$.

2.1.2 Classic Frequent Pattern Mining

For the inefficiency of the Apriori algorithm, Han et al. proposed a novel pattern structure called FP-Growth. As a divide-and-conquer strategy based algorithm, FP-Growth is comprised of two stages: the pre-processing stage and the mining stage. During the pre-processing stage, FP-Growth only scans the database $\mathcal{D}$ once to load all the 1-itemsets. Then the infrequent 1-itemsets are removed from the original database. Following this, a new database is retained and denoted as $\mathcal{D}'$. During the next procedure, the mining stage, an FP-Tree is built which refers to the database $\mathcal{D}'$. The FP-Tree is then regarded as several separate conditional databases, whereby each one is associated with one frequent pattern. In the following step, each database is mined iteratively, until no more candidates are generated. Finally, all of the frequent itemsets are discovered. Generally speaking, FP-Growth greatly reduces the time costs for scanning the database and generating candidates, thus it improves performance to a high level.

Regardless of the horizontal data format, where the database contains transaction-ID and transaction details (such details are comprised of a set of items), Zaki (Zaki 2000) proposed a novel vertical data format mining method called Eclat (Equivalence CLASS Transformation) for vertical data. In the vertical data format, a database is comprised of item and transaction-IDs list. The algorithm starts with 1-itemset after constructing the TID_set for each single item at the first scan of the database. In addition, the frequent (k+1)-itemset is generated from the frequent k-itemset due to the *Downward Closure Property*. The procedure is done after all frequent itemsets are computed and no more frequent itemsets or candidate itemsets can be generated.

2.1.3 Closed and Maximal Frequent Itemset Mining

One problem in mining frequent patterns/itemsets is to face the situation that a huge number of candidate patterns which have satisfied the *min_sup* threshold will be generated when mining in a quite large database. This is

also a great challenge to researchers especially when the threshold is set too low. Due to the Downward Closure Property, all subsets of one frequent itemset should also be frequent. Thinking about one long tail frequent itemset when the threshold set is really low, exponential numbers of redundant candidates would be mined during the mining procedure, which are smaller and frequent sub-patterns. The concepts of "*Closed Frequent Pattern Mining*" and "*Maximal Frequent Pattern Mining*" have been proposed to overcome such issues.

Here we have a definition of pattern P_a and P_b. A pattern P_a is said to be a *closed frequent pattern* in a dataset D if P_a is frequent in D and there exists no more super-pattern P_bs and that P_b has the same support as P_a in D. Also, if a pattern P_a is frequent, and there is no super-pattern P_b which $P_a \subset P_b$ is frequent in the dataset D, then the pattern P_a is called the *maximal frequent pattern* or *max-pattern*. For example, if itemset {a, b, c} is a closed frequent itemset, there should be no super-itemsets of it holds the same support as that of {a, b, c}. In addition, if this itemset {a, b, c} is a maximal frequent itemset, there should be no super-itemset of it whose support reaches the minimum support threshold.

Pasquier et al. are the first to propose the concept of mining closed frequent itemsets in 1999 (Pasquier, Bastide, Taouil & Lakhal 1999), where an Apriori-based approach algorithm called A-Close was introduced at the same time for such mining. The closed itemset lattice is defined by using a closure mechanism based on the Galois connection and Galois lattice theory (Birkhoff 1967). Another algorithm called CLOSET was proposed by Pei et al. in 2000 (Pei, Han, Mao et al. 2000) for mining closed frequent itemsets without candidate generation. This is a pattern-growth approach algorithm based on an FP-Tree (Han et al. 2000) structure, and a single prefix path compression method is applied for quickly identifying the closed frequent itemsets.

The major challenge in mining closed and maximal frequent itemset/pattern is to check whether a pattern is real closed or maximal. Due to the existing

papers, there are two strategies in tackling this problem:

- To keep track of the TID list of a pattern and index the pattern by hashing its TID values;
- To maintain the discovered patterns in a pattern-tree similar to FP-Tree.

Also, we have selected some representative algorithms to demonstrate such strategies.

- The first strategy is mentioned by Zaki and Hsiao (Zaki & Hsiao 2002) in their algorithm CHARM, which maintains a compact TID list called a "diffset". The CHARM simultaneously explores not only the itemset space, but also the transaction space, and thereby avoids enumerating all possible subsets of a closed itemset when enumerating the closed frequent itemsets.
- The second strategy is exploited by CLOSET+ algorithm by Wang et al. (Wang, Han & Pei 2003) as an extension work based on CLOSET (Pei et al. 2000). It is a depth-first search and horizontal format-based method which computes the local frequent items of a prefix by building and scanning its projected database. Further work such as FPClose (Grahne & Zhu 2003), which is another extension work based on FP-Growth, was first proposed by Grahne and Zhu in 2003. The main contribution of FPClose is to present a novel array-based technique that greatly reduces the need to traverse FP-Tree based algorithms and works even better in sparse datasets. In addtion, AFOPT (Liu, Lu, Yu, Wang & Xiao 2003) is also one outstanding algorithm in solving mining closed frequent pattern issue.

Bayardo was the first one to propose the algorithm for mining maximal frequent patterns (Bayardo Jr 1998). MaxMiner algorithm, an Apriori-based,

level-wise, breadth-first search approach was introduced to select those maximal frequent itemsets. In addition, two pruning strategies, super-itemset frequency pruning and sub-itemset frequency pruning, are applied in this algorithm for reducing search space. In addition, MAFIA was proposed by Burdick et al. in 2001 (Burdick, Calimlim & Gehrke 2001), where the vertical bitmaps were used to compress the transaction-ID list in order to increase the counting efficiency.

To compare with maximal frequent pattern, the closed frequent pattern is a lossless way to represent frequent patterns, which means it contains the complete information according to its original frequent patterns, whereas the maximal one usually does not contain the complete support information. Generally speaking, mining closed frequent itemsets means discovering all the closed patterns whose support is larger than the threshold ξ in database D.

2.1.4 Top-k Frequent Itemset Mining

Mining frequent patterns based on minimum support is sometimes quite difficult because of the lacking of database details. If the threshold is set to high, quite a few patterns will be discovered in the end. On the contrary, if the threshold is set quite low, a large quantity of candidates will be generated and the execution time will be very long. A range of factors will affect this, such as the density of the database, the length of a transaction, and the distribution of items. However, if one algorithm can only generate the given number of highest support patterns, which are also called, the top-k frequent patterns, such problems will no longer bother the users. Cheung and Fu proposed an algorithm called LOOPBACK and BOMO(Cheung & Fu 2004) to discover the N most interesting (high support) k-itemsets with the item-length $1 \leq k \leq k_max$, where k_max is a given value. The ExMining was proposed in 2006 (Quang, Oyanagi & Yamazaki 2006) to select top-k frequent itemsets via a two-phase mining method, which contains the "explorative mining" phase and the "actual mining" phase.

Take both closed method and top-k measurement, we have got the top-k closed pattern/itemset mining algorithms. Han et al. proposed such an algorithm TFP without the minimum support threshold in 2002 (Han, Wang, Lu & Tzvetkov 2002). The TFP starts with a preset minimum support 0, and this minimum support rises quickly by applying the properties of the top-k and the closed frequent itemsets as well as the length constraint.

2.1.5 Frequent Sequential Pattern Mining

To a market manager, the chronological order of transactions might not be necessary because the preference of the variety of people is not the same. However, to one person, some products might always be on his shopping list. He therefore chooses one specific brand instead of others for the majority of the time. Thus it is really useful for managers to know customer preference and make personalized plans in order to attract more customers and make profits. Such a database is called the sequence database, which is common and widely used in the customer shopping basket, web click-streams, biological gene sequence etc. To discover the frequent subsequence as patterns in a sequence database is the main idea of frequent sequential pattern mining. The sequence of one database is comprised of several ordered elements, and one element is an itemset (can be either a single item or more items).

Let $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ be a set of finite and discriminative items. A sequence is defined as $s = \langle e_1, e_2, ..., e_l \rangle$ where $e_j \subseteq \mathcal{I}, 1 \leq j \leq l$. A sequence database is defined as $D = \{[SID_1, S_1], [SID_2, S_2], ..., [SID_m, S_m]\}$. Here, an SID is a unique identification of the referred sequence. In addition, each sequence S is composed of several transaction itemsets with unique identification TIDs. For example, we still assume the itemset $\mathcal{I}$ is the items sold in some retail stores in Table 2.2.

The sequence database in Table 2.2 contains three sequences, which shows the historical shopping baskets for three customers. Sequence $SID = 1$ consists of 2 itemsets(transactions), and both $SID = 2$ and $SID = 3$ consists of 3 itemsets.

Table 2.2: Sequential Transaction Database

SID	TID	Transaction
S_1	T_1	bread, milk
S_1	T_2	bread, milk, butter, cheese
S_2	T_1	bread, butter, cheese
S_2	T_2	milk, butter, cheese
S_2	T_3	bread, cheese
S_3	T_1	bread, milk, cheese
S_3	T_2	milk, cheese
S_3	T_3	bread, milk, butter, cheese

$$s_1 = \langle (bread, milk)(bread, milk, butter, cheese) \rangle$$
$$s_2 = \langle (bread, butter, cheese)(milk, butter, cheese)(bread, cheese) \rangle$$
$$s_3 = \langle (bread, milk, cheese)(milk, cheese)(bread, milk, butter, cheese) \rangle$$

Referring to the inclusive and containment features between sequence and subsequence. A sequence $\alpha = \langle a_1, a_2, ..., a_p \rangle$ is a subsequence of sequence $\beta = \langle b_1, b_2, ..., b_q \rangle$, if and only if $\exists k_1, k_2, ..., k_p$, such that $1 \leq k_1 < k_2 < ... < k_p \leq q$ and $a_1 \subseteq b_{k_1}$, $a_2 \subseteq b_{k_2}$, ..., $a_p \subseteq b_{k_p}$, and denoted as $\alpha \subseteq \beta$. β is also super-sequence of α, or β contains α. For example, $\langle (bread, milk) cheese \rangle$ can be a subsequence of $\langle (bread, milk)(milk, cheese) \rangle$, which also means $\langle (bread, milk) cheese \rangle$ can be a subsequence of s_1 as well as s_3, but not s_2.

Numerous algorithms to tackle the sequence pattern mining problem have

been proposed since it was first mentioned in (Agrawal & Srikant 1995). At the same time Agrawal and Srikant introduced the concept of sequence, they proposed a new algorithm called AprioriAll (Agrawal & Srikant 1995), which is widely regarded as the first level-wise algorithm in solving sequential pattern mining issues. Just as the name implies, AprioriAll is the "Apriori" algorithm in mining sequences. First, it finds all frequent 1-patterns (the patterns with one item) whose support is larger than the user-specified threshold. Two kinds of list containers named candidate lists and frequent pattern lists are initialised and maintained in the second step. Finally, the following process repeats until no more candidates can be found: for each (k + 1)-candidate frequent pattern that is merged by two frequent k-patterns, the support is scanned from the original database in order to satisfy the minimum support.

In 1996, Srikant and Agrawal proposed an advanced sequential pattern mining method called GSP(Generalized Sequential Patterns) (Srikant & Agrawal 1996). This is also an extension of Apriori and the basic structure is similar to AprioriAll. However, the candidate generation and candidate support counting methods may be viewed as two major differences. For the candidate generation process, a mechanism is applied for pruning the unpromising candidates(those whose support is less than the threshold). Thus the amount of candidates reduces substantially compared to AprioriAll for the same length of candidate patterns. For the candidate support counting process, a hash-tree is introduced in order to reduce the number of candidates to be checked during the scanning.

In addition, SPADE (Zaki 2001) is proposed for a vertical format-based sequential pattern mining method, which builds an ID-list structure for each candidate. SPADE is also an extension of several former vertical format-based frequent pattern mining models, like Eclat (Zaki 2000) and CHARM (Zaki & Hsiao 2002). PrefixSpan, a pattern-growth approach to sequential pattern mining, was proposed by Pei et al. (Pei, Han, Mortazavi-Asl, Pinto, Chen, Dayal & Hsu 2001) which works in a divide-and-conquer way. Based on performance, PrefixSpan works better than the algorithms mentioned before.

2.1.6 Weighted Frequent Itemset Mining

During the process of FIM increasing and marketing requirement, the only measurement of support or the only counting of category could not satisfy the business and financial needs, thus a new perspective should take into consideration the weighted value of each item in the database, also called weighted frequent itemset mining. To arrange the weighted value of each item into frequent itemset mining is more practical because it considers the weighting attribute significance (weight) of different items, which is ignored in the traditional frequency/support framework.

Let $D = \{T_1, T_2, ..., T_m\}$ be a transaction database, and $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ be a set of finite and discriminative items. Each transaction $T_c \in D$ $(1 < c < m)$ is a subset of $\mathcal{I}$. Each item in $\mathcal{I}$ is associated with a weighted value to form a weighted item, denoted as $w(i_k)$, and the value is called *Item Weight*. Then the item is paired with its weighted value, denoted as $\langle i_k, w(i_k)\rangle (1 \leq k \leq n)$ in transactions.

The *Itemset Weight* of an itemset X is derived from the weights of its enclosing items, denoted as w(X) and defined as

$$w(X) = \frac{\sum_{i_k \in X} w(i_k)}{l} \tag{2.2}$$

Here, l is the length of itemset X. In addition, the *Transaction Weight* is also a kind of itemset weight and attached to each of the transactions. Thus, the weighted support of an itemset X is denoted as $wsup(X)$ and defined as

$$wsup(X) = w(X) \times supp(X) \tag{2.3}$$

Here, $supp(X)$ is the traditional support of itemset X. Also, an itemset X is called a weighted frequent itemset if its $wsup(X)$ is no less than a user-specific weighted threshold.

Some early related work such as MINWAL (Cai et al. 1998), WAR (Wang et al. 2000) and WARM (Tao et al. 2003) are extensions of Apriori-based approaches in mining weighted frequent itemset mining. However, the dis-

advantage of such a level-wise algorithm is the multiple database scanning, which makes the performance really poor. Accordingly, a novel technique FP-Growth-based approach is applied for overcoming this task in the later work, such as the WFIM algorithm in 2005 (Yun & Leggett 2005*a*).

In addition, Yun and his group have designed some efficient algorithms in weighted frequent itemset mining. The novelty of WLPMiner is the application of length decremental support constraints (Yun & Leggett 2005*b*, Yun 2008*a*). The WIP algorithm discovers weighted interesting patterns which not only takes into consideration a balance between the two weighted and support measurements, but also the strong weight and/or support affinity into consideration (Yun & Leggett 2006*a*, Yun 2007*a*), thus fewer but more valuable patterns can be generated. Some hybrid algorithms which apply other constrains also make the issue more interesting, and the algorithm itself more reliable. WCloset extracts lossless closed weighted frequent pattern (Yun 2007*b*), and WSpan discovers the weighted frequent sequential pattern (Yun & Leggett 2006*b*, Yun 2008*b*).

Other novel structures for mining weighted frequent itemsets/patterns of variety domains have been proposed in recent years. A single-pass structure of database to capture the weighted interesting patterns called the SPWIP-tree structure was introduced by Tanbeer et al. in 2008 (Tanbeer, Ahmed, Jeong & Lee 2008), later two algrithms $IWFP_{WA}$ and $IWFP_{FD}$ were proposed as an improvement of the WFP miner (Ahmed, Tanbeer, Jeong, Lee & Choi 2012). Another new framework for catching time-interval weighted sequential (TiWS) patterns in a sequence database and a new concept of time-interval weighted support (TiW-support) are used to find the TiWS patterns (Chang 2011). For mining weighted frequent sub-graphs with weight and support affinities, MWSA is proposed by Lee and Yun in 2012 (Lee & Yun 2012). MWFIM is proposed for mining maximal weighted frequent patterns in 2012 (Yun, Shin, Ryu & Yoon 2012).

2.2 High Utility Pattern Mining

In this section, we will give an overview of the utility framework proposed in the literature. Since the majority of applications of the utility model are in the marketing and business fields, we will make such models the focus of the section. We firstly summarise utility definitions followed by the various fast algorithms as well as the utility frameworks.

2.2.1 The Utility Framework

Large numbers of papers have been proposed for mining frequent patterns. These patterns contain various types of data, such as itemsets, constrained rules, graphs, sequences and so on. So far, such algorithms are only constructed for patterns with high support or frequency, and some consider association at most. In the support-confidence framework, patterns whose support or/and confidence is less than the minimum threshold are considered to be a waste and even the evaluation of algorithms considers how fast those useless patterns can be pruned. However, when frequency is regarded as the only interestingness measurement, and all the items are treated equally in transactions, the results are not always reliable and useful in handling some real world problems. For one counter-example, this frequency-association assumption is contradicted when the importance or the interestingness of different items/itemsets/graphs/sequences varies significantly and thus this frequency framework loses its adaptiveness in this circumstance. Researchers proposed the utility-based framework in order to solve the above conflict. We briefly introduce some definitions in utility framework as follows.

Definition 2.2 *Let* $D = \{T_1, T_2, ..., T_m\}$ *be a transaction database, and* $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ *be a set of finite and discriminative items. The set of items in each transaction* $T_c \in D$ $(1 < c < m)$ *is a subset of* $\mathcal{I}$*. In addition, each transaction appears with a distinct identifier called* TID*. In a given transaction* T_c*, each item* i_k *appearing with a positive integer, denoted as* $q(i_k, T_c)$ *is called* i_k*'s* quantity utility *or* i_k*'s* external utility *in* T_c*. Thus a*

Table 2.3: Utility Dataset

TID	Transaction	TU
T_1	(A, 1) (C, 1) (D, 1)	8
T_2	(A, 1) (B, 6) (C, 2) (F, 5)	24
T_3	(A, 2) (B, 2) (C, 6) (D, 5) (E, 1)	60
T_4	(B, 4) (C, 3) (D, 2) (F, 3)	18
T_5	(B, 2) (C, 2) (F, 3)	9

Table 2.4: Profit Table

Item	A	B	C	D	E	F
Profit	5	2	1	2	30	1

transaction with four items can be presented as

$$T_c = (i_1, q(i_1, T_c))(i_2, q(i_2, T_c))(i_3, q(i_3, T_c))(i_4, q(i_4, T_c)) \tag{2.4}$$

Also, each item in $\mathcal{I}$ is associated with a positive number $p(i_k, \mathcal{I})$, which is called i_k's profit utility *or i_k's* internal utility *in $\mathcal{I}$.*

A simple example of the utility dataset is shown in Table 2.3 and Table 2.4. Specifically, the utility dataset table contains TID, which is a unique identification for each transaction. Transaction items have their quantity utility and TU, which is short for transaction utility. The profit table contains each item with their profit utility.

Definition 2.3 *The* utility of item i_k *in a transaction T_c is the profit utility*

of the item times its quantity utility in a transaction, defined as

$$u(i_k, T_c) = p(i_k, \mathcal{I}) * q(i_k, T_c) \tag{2.5}$$

An item i_k with its utility in a transaction T_c is denoted as $u(i_k, T_c)$ $(i_k \in T_c)$.

Definition 2.4 *The* utility of an itemset X *in a transaction* T_c *is the utility sum of all items belong to the itemset, defined as*

$$u(X, T_c) = \sum_{i_k \in X} u(i_k, T_c) \tag{2.6}$$

An itemset contains l discriminative items is called an *l-length* itemset, where $X \subseteq \mathcal{I}$. The utility of the same item in different transactions might be different considering the quantity of each item purchased.

Definition 2.5 *The* utility of an itemset X *in the whole database* D *is the sum of the utility of this itemset in all transactions. It is denoted as* $U(X)$*, and defined as*

$$U(X) = \sum_{X \subseteq T_c \wedge T_c \in D} u(X, T_c) \tag{2.7}$$

In addition, one item can be regarded as an 1-length itemset.

Definition 2.6 *The minimum utility threshold is denoted as* min_util*, and a set of all itemsets whose utilities are higher than* min_util *is denoted as* $f_H(D, min\_util)$*. The goal of HUI mining is to find such itemset,* $f_H(D, min\_util)$*.*

Definition 2.7 *The* transaction utility *of the transaction* T_c *is denoted as* $TU(T_c)$ *and defined as*

$$TU(T_c) = \sum_{i_k \in T_c} u(i_k, T_c) \tag{2.8}$$

Definition 2.8 *The* transaction-weighted utility *of the itemset* X *is the sum of the transaction utilities of all the transactions that* X *belongs to. It is*

denoted as $TWU(X)$ *and defined as*

$$TWU(X) = \sum_{X \subseteq T_c \wedge T_c \in D} TU(T_c) \tag{2.9}$$

Definition 2.9 *The* high transaction-weighted utility itemset (HTWUI) *consists of those itemsets whose TWU is no less than* min_util.

Property 2.2 *The* transaction-weighted downward closure property *holds for HUI, says that if an itemset* X *is not a HTWUI, all its supersets are not HUIs because* $U(X) \leq TWU(X)$.

The utility framework can be regarded as an extension of frequency/support framework. The main difference is that in utility framework, each item is associated with two utility values. One, presented in the record of transaction, is called its quantity or external utility; the other, presented in an individual list, is called its profit or internal utility. The advantage of such additional property is to change the equivalent of two different items and increase the weight of those items with higher interestingness. Let us take a basket of apples and oranges as an example. In the frequency framework, we only have two items: apple and orange; whereas, according to the utility framework, we need to involve additional attributes such as the volume of each fruit and the price of each fruit.

Researchers have been working on utility mining for more than ten years and are still working on that. At the beginning of this section, we give a brief overview of some existing outstanding papers as well as the algorithms in high utility data mining domain. we separate those papers into two parts: itemset mining and sequence mining. each part contains several interesting topics, and for each topic, the algorithms are ordered by their publish time.

High utility itemset mining

There are three topics in high utility itemset mining.

- Mining high utility pattern. The term of "mining high utility itemsets" was first introduced in 2003 by Chan et al. (Chan, Yang & Shen 2003),

and the algorithm was named OOApriopi. However, the concepts and definitions in this paper are not how they are actually defined today. One year later, Yao et al. (Yao et al. 2004) proposed a foundational approach for mining high utility itemsets, this algorithm is the so-called foundational, and it is also referred to as MEU in several later works. In 2005, the Two-Phase was proposed, which was the major control article and was widely cited in most of the work published between 2006-2010 (Liu et al. 2005). Several papers came out from 2007 to 2010 to demonstrate various approaches for mining high utility patterns, such as HUQA (Yen & Lee 2007) by Yen and Lee, CTU-Miner (Erwin, Gopalan & Achuthan 2007*b*) by Erwin et al., HUPTPM (Zhou, Liu, Wang & Shi 2007) by Zhou et al. all in 2007, IIDS (Li, Yeh & Chang 2008) by Li et al., CTU-Prol (Erwin, Gopalan & Achuthan 2008) by Erwin et al. as well in 2008, and HUC-Prune (Ahmed et al. 2009) by Ahmed in 2009. In 2010, a novel tree structure algorithm for mining utility itemsets with efficiency and accuracy named UP-Growth was proposed by Tseng et al. (Tseng, Wu, Shie & Yu 2010). It comes to be treated as a milestone in the utility mining domain. In 2010s, several algorithms were proposed. CHUD was proposed by Wu et al.(Wu et al. 2011) in 2011, d^2HUP by Liu et al. (Liu et al. 2012), HUIM by Liu and Qu (Liu & Qu 2012), TKU by Wu et al. (Wu et al. 2012), Udepth by Song et al. (Song, Liu & Li 2012) in 2012, FHM by Fournier-Viger et al. (Fournier-Viger, Wu, Zida & Tseng 2014), and CHUI-Mine by Song et al. (Song, Liu & Li 2014).

- Mining incremental high utility pattern. IHUP (Ahmed et al. 2009) by Ahmed in 2009 was the first algorithm for mining incremental high utility itemsets. One year later, Lin et al. also proposed an incremental algorithm (Lin, Hong, Lan, Chen & Kao 2010, Lin, Lan & Hong 2012) published in 2010 and 2012. In 2013, Lin's group proposed another incremental algorithm for mining high utility patterns based on the pre-large concept.

- Mining high utility patterns in data stream. As early as 2006, Tseng et al. proposed the issue of mining high utility patterns in the data stream. THUI-Mine was put forward to tackle this topic at the same time (Tseng, Chu & Liang 2006). In 2008, Li's group began to examine this problem and proposed an algorithm called MHUI-TID for fast and memory efficient mining (Li, Huang, Chen, Liu & Lee 2008). In 2011, negative item profit was also taken into consideration in this algorithm (Li, Huang & Lee 2011). In 2010, two algorithms GUIDE and HUPMS were proposed by different groups Shie et al. (Shie, Tseng & Yu 2010) and Ahmed et al. (Ahmed, Tanbeer & Jeong 2010*a*), and two years later, their journal article versions were published (Shie, Philip & Tseng 2012, Ahmed, Tanbeer, Jeong & Choi 2012)

High utility sequential pattern mining

After the sequential concept was proposed in the pattern mining domain, Ahmed and his group proposed a novel approach for mining high utility patterns in a sequential database in 2010 (Ahmed, Tanbeer & Jeong 2010*c*), whereby the algorithms UL (UtilityLevel) and US (UtilitySpan) could be employed to successfully discover high-utility sequential patterns successfully. The US performed better than UL with less candidates generated. In 2012, Yin et al. proposed a much more efficient algorithm being the LQS-Tree (Lexicographic Q-Sequence Tree) (Yin et al. 2012). Because of the application of a depth-first recursively invoked procedure, this algorithm presented good outcomes in both the synthetic and real datasets for detecting high utility sequences from large scaled data even if they possessed a very low minimum utility threshold. One branch topic of frequent sequential pattern mining is frequent episode mining, which was first proposed by Mannila et al. in 1995 (Mannila, Toivonen & Verkamo 1995). Here, an episode is defined to be a collection of events that occur within time intervals of a given size in a given partial order. To introduce the concept of episode in to utility sequential pattern mining, Wu et al. (Wu, Lin, Yu & Tseng 2013) proposed

an algorithm called UP-Span to discover the high utility episodes in complex event sequences. This algorithm focuses on handling two major issues: in one, the assumption of useful information-utility is considered; in the other, the episodes in complex event sequences instead of simple sequential events are addressed. Considering the top-k mining methods on frequent pattern and high utility mining, Yin et al. proposed a closed approach for mining top-k high utility sequential patterns in 2003 (Yin, Zheng, Cao, Song & Wei 2013). These regard the DCP (Downward Closure Property) as a fundamental property.

2.2.2 Fast Algorithms for Mining HUIs

In this subsection, we provide an overview of state-of-the-art algorithms in HUI Mining.

Foundational (MEU) Algorithm

In 2004, Yao et al. proposed a foundational approach to mining itemset utilities (Yao et al. 2004), which is widely believed to be the first foundational paper for mining high utility itemsets. Accordingly, the algorithm was called "Foundational" or "MEU". The issue of discovering high utility itemsets was defined in this paper. In addition, the theoretical concept and model of mining itemset utilities were proposed as well. Also two types of utilities for items were defined. These were named *transaction utility* (which is the *internal utility* above) and *external utility*. We say this paper is the foundational one because some of the definitions are still used today. For example, the *utility of an item in a transaction*, the *local utility of an item in an itemset* (which is the utility of an item in database above), and the *utility of an itemset*.

Yao was the first to point out that the DCP (Downward Closure Property) for mining frequent patterns is no longer applicable for mining high utility itemsets. When calculating the utility of an itemset, the itemset utility can be either increase or decrease as the itemset is extended by adding

items. It is necessary to discover new properties of itemset utility to make the efficient algorithms possible. In order to solve this problem, the *Utility Bound Property* and the *Support Bound Property* are proposed. In plain words, the utility bound property describes that the utility of a k-itemset I^k must be less than or equal to the sum of all its (k-1)-sub-itemset utilities. The support bound property demonstrates that the support of a k-itemset I^k must be less than or equal to any one of its (k-1)-sub-itemset, which is the same meaning as the DCP. In addition, a heuristic model to predict the *expected utility value* is proposed based on two properties. The expected utility value is defined as below

$$u'(I^k) = \frac{supmin}{k-1} \sum_{i=1}^{k} \frac{u(I_i^{k-1})}{sup(I_i^{k-1})} \tag{2.10}$$

where,

$$supmin = \min_{\forall I_{i_p}^{k-1} \subset I^k} \{sup(I_{i_p}^{k-1})\} \tag{2.11}$$

The notations in the equations above are explained as follows. $u'(I^k)$ is the expected utility value of a length-k itemset I^k. The $supmin(I^k)$ is the minimum of all support values of I^k's (k-1)-sub-itemsets.

For each k-itemset I^k, the number of (k-1)-sub-itemsets is k. Among all k itemsets' supports , the $supmin(I^k)$ is selected, thus the function is not very efficient. The authors proposed the *Utility upper bound property* as well as two pruning strategies in their later journal work (Yao & Hamilton 2006) to exclude unpromising candidates.

Two-Phase Algorithm

As referred in Yao's paper (Yao et al. 2004), the DCP is no longer applicable, and a new property should be found for utility mining. Liu et al. published the paper "A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets" one year later in Pacific-Asia Conference on Knowledge Discovery and

Table 2.5: TDCP VS DCP

Transaction-weighted Downward Closure Property	Downward Closure Property
High Utility Itemset Mining	Frequent Pattern Mining
$TWU(I^k) \leq TWU(I^{k-1})$	$sup(I^k) \leq sup(I^{k-1})$

Data Mining(PAKDD)'2005. This paper is one of the most cited papers with its algorithm called Two-Phase or TP in the utility mining domain.

The main contribution of Two-Phase is the definition of the *Transaction-weighted Utilization (TWU)* and the *Transaction-weighted Downward Closure Property (TDCP)* based on TWU. The definitions of TWU and TU are referred to at the beginning of this section. The Two-Phase algorithm consists of two phases, just as its name suggests. In the first phase, the algorithm generates a set of high utility candidates, all of the *high transaction-weighted utilization itemsets (HTWUIs)* are in this set including the high utility itemsets. Then in the second phase, all high utility itemsets are filtered from the set of high transaction-weighted utilization itemsets after one database scans.

The transaction-weighted downward utility property is such an important property that it introduces the downward closure property into the utility domain perfectly. It has also been widely used in later works. The TDCP is much like DCP in traditional frequent pattern mining. The property is quite succinct in description, compared with DCP for frequent itemset mining and it is described in Table 2.5.

In addition, the HTWU is supposed to be the collection of all high transaction-weighted utilization itemsets, whose TWU is no less than a user specified threshold ξ' in a transaction database. Also, the HU is supposed to be a set of all high utility itemsets, whose utility is no less than another user given threshold ξ. Thus it is not hard to prove $HU \subseteq HTWU$ when $\xi = \xi'$. As to the Two-Phase algorithm, HTWUs are generated in Phase 1, and HUs

are selected in Phase 2.

In short, the TWU mining model outperforms the MEU model by several aspects. The TWU of I^k is independent from that of I^{k-1}, and the upper bound of TWU is tighter than the expected utility in MEU, thus fewer candidates are generated and less time is cost. In the second place, it is more accurate to apply the utility measurement instead of the support measurement as the upper bound for generating candidates. Finally, the TWU model in Two-Phase which only incurs the add operation is less complex in terms of arithmetic than the multiplications in MEU.

Isolated Items Discarding Strategy

In 2007, Li et al. proposed a strategy called Isolated Items Discarding Strategy (IIDS). This can be applied to all existing level-wise (Apriori-based) utility mining methods, especially to the models for share mining algorithms which also work well for utility mining. It works in an efficient way via reducing the number of candidates, thus performance improves. For the most efficient known share mining models being the Share-counted Fast Share Measure (ShFSM) (Li & Yeh 2005) and the Direct Candidates Generation (DCG) (Li, Yeh & Chang 2005), new algorithms called Fast Utility Mining (FUM) and DCG+ were proposed and these proved to be extremely good for utility mining.

The IIDS strategy shares some similarities with the TWU (Transaction Weighted Utility) strategy but it has a tighter upper bound in relation to identifying and abandoning isolated items from transactions. Unlike the TWU process which only scans the database once to obtain all of the items' TWU and discards the low TWU items, the IIDS process scans the database many times. The initial candidate set is composed of all the items from the original database. The candidate set is smaller after each scanning because the unpromising items are removed from the set. The strategy terminates when no more isolated items can be removed and the minimised candidate set is generated.

In general, all the algorithms demonstrated above are based on the level-wise approach, called the generation-and-text approach, which was first demonstrated in 1994 (Agrawal et al. 1994).

CTU-Mine, CTU-PRO & CTU-PROL

Erwin et al. proposed a series of algorithms where the Compressed Transaction Utility (CTU) strategy could be applied. This strategy allows a compact data representation structure named the CTU-tree, which comes from the basic CT-tree (Sucahyo & Gopalan 2003), to discover high utility itemsets efficiently. With the application of such structures, algorithms for utility mining using the pattern growth approach instead of the candidate generation-and-text approach are proposed for mining dense data with the CTU-Mine (Erwin et al. 2007*b*). In addition, a fixed compact data representation named the Compressed Utility Pattern Tree (CUP-Tree) is developed for mining a sparse data structure with the algorithm called CTU-PRO (Erwin, Gopalan & Achuthan 2007*a*). Moreover, the CTU strategy is also applied for mining subdivisions from parallel projections independently while the algorithm called CTU-PROL (Erwin et al. 2008) can be used for mining high utility itemsets from datasets that are too large to be held in the main memory. One advantage of the pattern growth approach rather than the level-wise approach is that it skips the second phase of scanning the database a second time to achieve the real high utility itemsets. Below are the details of these three algorithms as well as the related structures.

CTU-Mine (Erwin et al. 2007*b*) proposed the first of three algorithms with a data structure called the CTU-tree which is designed for representing the transaction data required for compact utility mining and it is derived from the basic CT-Tree, a kind of compressed prefix tree. The CTU-tree contains two parts: ItemTable and Compressed Transaction Utility Tree. The ItemTable consists of six fields, including the index or item-id of each universal item, original item-id, profit per unit of the item, quantity of each item, TWU and a pointer to the root of subtree. On the other hand, the

CTU-tree compresses the quantities of transactions as well as all transactions of high TWU items. In addition, each node from the CTU-tree contains an index and is associated with an array of TWU for the patterns at that node. The roof of the tree is the first item in the ItemTable.

The construction of ItemTable comes first and then the CTU-tree is built. The mining process of the CTU-Mine is the process of a pattern-growth CTU-Tree construction, where a roof is first selected and all subtrees are inserted after scanning the database once. All relevant patterns and related information are inserted into the CTU-tree one by one with the scanning of each transaction, which is also the way a CTU-tree is reconstructed. Thus the TWU of the existing node is added or a new node is generated after the end of the current related node has occurred when a new transaction is scanned.

The performance of the CTU-Mine is efficient in dense data, but quite poor in terms of the sparse data. To tackle this issue, Erwin et al. proposed a bottom-up projection based algorithm called CTU-PRO (Erwin et al. 2007*a*). The CTU-PRO algorithm also traverses a Compressed Utility Pattern Tree (CUP-tree), which is quite similar to the CTU-tree in the CTU-Mine, for mining high utility itemsets. A major enhancement of the CUP-tree is that a special link is applied between the same items or nodes with the same item-id. This makes the CUP-tree similar to the FP-tree(Han et al. 2000) for frequent pattern mining. When new transactions are scanned and new items are added to the current pattern, these special links help to directly locate the next target node without rescanning the whole tree. The main process is as follows: The CUP-tree is generated by CTU-PRO from the transaction database after the first scan. This tree is called the global CUP-tree which contains all identified individual high TWU items. A local CUP-tree which is created as a projection from each high TWU items is extracted from the global CUP-tree for mining all high utility patterns with the high TWU item as a prefix.

In addition, CTU-PROL(Erwin et al. 2008) as an extension of the CTU-

Mine and CTU-PRO was proposed by Erwin et al. particularly for mining datasets which are too large to be held in the main memory. By the application of parallel projections, the algorithm creates subdivisions that can subsequently be mined independently. In CTU-PRO, a global CUP-tree as well as local CUP-trees for individual items are applied, while in CTU-PROL, CUP-trees are used in each subdivision to obtain the complete set of high utility itemsets. In addition, the Transaction Weighted Downward Closure Property (TWDCP) is applied to reduce the search space for each subdivisions.

IHUP

The IHUP (Incremental High Utility Pattern) algorithm was proposed to handle the issue of high utility pattern mining from the incremental database with consideration for numbers of insertions, deletions, and modifications for currently available memory size by Ahmed (Ahmed et al. 2009) in 2009. An incremental database is more practically significant than a fixed database in marketing analytics and social economics. For example, the sale quantities and prices of some products in a supermarket might change constantly. Different promotion packages and different discount rates will affect both the volumes and prices or profits. In addition, one customer might get a refund when he/she finds that the price of the same product may be less at another time or in another store. This phenomenon leads to the database being rebuilt because of inserting, deleting or modifying transactions, thus the utility of each pattern is no longer applicable and the re-mining process of the updated database is quite time-consuming. IHUP is proposed to avoid such a re-mining process with the introduction of the "build one mine many" approach proposed in (Cheung & Zaiane 2003, Koh & Shieh 2004) for frequent pattern mining from incremental databases.

In the IHUP algorithm, three tree structures are proposed for efficient interactive and incremental database mining. The first tree structure which arranges the items lexicographic order and insets them as a branch inside

the tree, is called the Incremental HUP Lexicographic Tree ($IHUP_L - tree$). In addition, all the IHUP-trees explicitly maintain transaction weighted utility and transaction frequency values in both the header table and the tree nodes. The second tree structure is called the IHUP Transaction Frequency Tree ($IHUP_{TF} - tree$), where a compact size is obtained by arranging items according to their transaction frequency in descending order. This tree is applied to reduce the size of the $IHUP_L - tree$ with the prefix-sharing increment inside it. The last tree, called IHUP Transaction-Weighted-Utilization Tree ($IHUP_{TWU} - tree$), is proposed to reduce the execution time and it is designed based on the TWU value of items in descending order because several low-TWU items can appear before the high-TWU items in branches of both the $IHUP_L - tree$ and the $IHUP_{TF} - tree$.

FUP-HUI, FUP-HU and Pre-HUI

Other algorithms besides the IHUP were proposed by Lin et al. for mining high utility itemsets with incremental pattern structures. These were proposed between 2010 and 2014 (Lin, Hong, Lan, Chen & Kao 2010, Lin et al. 2012, Lin, Hong, Lan, Wong & Lin 2013, Lin, Hong, Lan, Wong & Lin 2014).

The FUP-HUI algorithm is based on a combination of two algorithms, one is a basic approach in HUI (High Utility Itemsets) mining called Two-Phase (Liu et al. 2005) and the other is an extension approach of frequent pattern mining called Fast-UPdate (FUP) (Cheung, Han, Ng & Wong 1996). In this algorithm, itemsets are partitioned into four parts depending on whether they are high transaction weighted utilisation itemsets in original databases or newly inserted transactions. Each part is then executed by its own procedure.

The Pre-HUI algorithm is based on two algorithms, Two-Phase and the pre-large concept for efficiently maintaining the discovered rules in incremental data mining proposed by Hong et al. (Hong, Wang & Tao 2001). This is an incremental mining algorithm which efficiently maintains, the discovered high utility itemsets based on the pre-large concept. It first partitions item-

sets into nine cases according to whether they are large (high), pre-large or small transaction-weighted-utilisation itemsets in the original database and in the inserted transactions. Each part is then executed by its own procedure. In addition, the Downward Closure Property (DCP) is applied to reduce the size of the candidates in order to decrease the computational time of scanning the database.

d^2HUI

In 2012, Liu et al. proposed another high utility itemset growth approach (Liu et al. 2012) that worked in a single phase without generating candidates. The main contribution of d^2HUI is listed as follows:

- A pattern growth based approach is proposed, which enumerates an itemset as a prefix extension of another itemset with powerful pruning, thus no itemset can be visited before its subset. This allows efficient computation of each enumerated itemset in order to directly identify high utility itemsets and to save search space. In addition, a tighter upper bound than that of TWU (Liu et al. 2005) is proposed.

- Different strategies are incorporated with this approach. For dense data, a lookahead strategy is applied, which tries to identify high utility itemsets earlier based on a closure property and a singleton property so that invalid and costly enumeration can be avoided. For spares data, a method of identifying and discarding some of the irrelevant items with a relatively loose basic upper bound is provided, thus the upper bound can be further tightened.

- A novel data structure is proposed to represent all the original utility information from raw data. A target to the root cause of candidate generation with the existing FP-growth based algorithms (Ahmed et al. 2009, Erwin et al. 2008, Tseng et al. 2010) makes it not only more efficient in the computation of utilities and upper bounds for enumer-

ated itemsets but also less costly in terms of memory space cost than the tree structures used by these algorithms.

These advantages allow the d^2HUI approach to enumerate itemsets by prefix extensions, to prune search space by a tighter utility upper bound, and to maintain original utility information in the mining process by a novel data structure.

UP-Growth and UP-Growth+

A novel structure for efficiently mining high utility itemsets was proposed by Tseng et al. (Tseng et al. 2010) in 2010. It is also a tree-based structure called the Utility-Pattern-Tree (UP-tree). However, compared with other existing tree structures such as the FP-tree based structure IHUP(Ahmed et al. 2009), which uses Transaction-Weighted-Utilisation to generate the high utility candidates, the UP-tree generates less candidates during the period of tree construction. This is because the IHUP generates the same number of candidates as the Two-Phase algorithm, which first proposed the concept of TWU and Transaction-weighted Downward Closure Property. However, this approach overestimated the utilities of candidates and too many high TWU candidates proved to be low utility itemsets in the end.

A refined tree structure with four efficient strategies is proposed to overcome this issue. A Global UP-tree is constructed with the following two strategies.

- The Discarding Global Unpromising (DGU) items strategy is the first strategy which has been applied for eliminating unpromising items and their utilities from the transaction utilities during the construction of a global UP-tree. This is because the unpromising items have no interestingness in relation to the high utility itemsets mining process.

- Discarding Global Node (DGN) utilities is the second strategy applied for global UP-tree construction. During this process, the utilities of all

the nodes' descendants are discarded from the utility of the node. Each node in a UP-tree is comprised of the name of the item, its support and its utility, which is also the estimated utility of the node. In addition, within a global tree, the node utility is the accumulated utility from the first item to the current node (item). Therefore, the DGN makes a tighter utility upper bound than other strategies using TWU.

Furthermore, two strategies are applied to build local Up-trees and to prune unpromising items.

- The Discarding Local Unpromising (DLU) items strategy is applied each time to build a local UP-tree. During this process, the items whose utility is lower than the minimum utility threshold are discarded from the utility path.
- The Decreasing Local Node (DLN) utilities strategy is applied as the last strategy to decrease the utility of each node whose descendant utilities are lower than the minimum utility threshold.

The algorithm UP-growth+ is an enhanced algorithm based on UP-growth (Tseng, Shie, Wu & Yu 2013). This structure is not used during the construction of the global UP-tree, but applied to estimate two tighter utility upper bounds for building the local UP-trees. Two renewed strategies named DNU and DNN are proposed and these have proved to be more efficient based on experiments.

TKU

Basic pattern mining algorithms refer to the discovery of itemsets where the estimated value (frequency, utility, and so on) is higher than a user-specified minimum value threshold. However, even though numerous studies have been proposed in this domain, users are still unable to set an appropriate threshold. Actually, a reasonable threshold strongly influences the results and execution time. If this minimum threshold is set too low, lots of candidates

are generated and the algorithm is inefficiency because of the execution time, or the memory even runs out. On the other hand, if the minimum threshold is set too high, few patterns are discovered. The $Top-k$ approach is proposed to tackle this issue. For high utility itemsets mining, TKU (Top-k Utility), which is proposed by Wu et al. in 2012 (Wu et al. 2012), exploits a fixed size candidate space to maintain the high utility candidates since the minimum threshold is not given at first. When this space is not full, the threshold can be regarded as 0, otherwise the threshold is the least utility in the space.

Just as the CHUD (Wu et al. 2011), TKU exploits the UP-tree and it is an extension from UP-growth. In addition, five efficient strategies are proposed to raise the minimum threshold during different stages of the mining process.

- The first strategy is called MC and it is applied to raise the threshold by MIU of the newly mined PKHUI if the itemset's MIU, TWU, MAU are no less than the current threshold. Here, MC, PKHUI, MIU, TWU, MAU are short for MIU of Candidate, Potential top-K High Utility Itemset, Minimun Item Utility, Transaction Weighted Utilisation, and Maximal Utility. The MIU of an itemset is the sum of all MIUs of the items in this itemset times the support of this itemset.

- The second strategy is called Pre-Evaluation (PE) and this is applied during the first scan of the database to insert all the items' utilities into the pre-evaluation matrix.

- The third strategy is called Raising the threshold by Node Utilities (NU) and this is applied during the construction of the Up-tree, which is also the second scan of the database. The DGN (Discarding Global Node) strategy in the UP-tree (Tseng et al. 2010) guarantees that the utility of the candidate is higher than the node utility, thus the threshold is raised if it is less than the k-th highest node utility.

- The fourth strategy is called Raising the threshold by MIU of Descendents (MD) and it is applied after the construction of the UP-tree and

before the generation of PKHUIs. For each node N_α under the root of Up-tree, this strategy first calculates the support count of $N_\alpha N_\beta$ by traversing every descendent node N_β of N_α.

- The last strategy is called Sorting candidates & raising threshold by the exact utility of candidates (SE) and this is applied during Phase Two of TKU. In this strategy, all the candidates generated in Phase One are sorted in descending order of their estimated utilities. If there are more than k HUIs to satisfy the minimum threshold, the threshold can then be raised to the k-th highest exact utility.

2.2.3 High Utility Sequential Pattern Mining and High Utility Episodes Mining

High Utility Sequential Mining Framework

Frequent Sequential Pattern Mining (FSPM), as an extension study of frequent pattern mining, was first proposed by Agrawal et al. in 1995 (Agrawal & Srikant 1995). Since the FSPM had been extensively studied, and the utility framework had been widely applied, the incorporation of the utility concept into sequential pattern mining began. It is widely believed that Ahmed et al. were the first to bring the utility framework to the sequential pattern mining (Ahmed et al. 2010*c*) as well as web access mining (Ahmed, Tanbeer & Jeong 2010*b*). Even though Zhou et al. referred to this web log sequence mining topic earlier in 2007 (Zhou et al. 2007), his approach is a simple and straightforward application of the Two-Phase (Liu et al. 2005). Yin et al. proposed efficient algorithms for high utility sequential pattern mining in 2012 (Yin et al. 2012) and mining top-k high utility sequential patterns in 2013 (Yin et al. 2013). Lan et al. introduced the fuzzy concept into the framework in 2013 (Lan, Hong, Huang & Pan 2013) and proposed the maximum (Lan, Hong & Chao 2014*a*), minimum (Lan, Hong & Chao 2014*b*) utility constraints measurement in 2014. Wu et al. first proposed high utility episodes mining in 2013 (Wu et al. 2013).

The basic concept of high utility sequential pattern mining is a combination of the frequency-based sequential pattern mining and high utility itemsets mining. Specifically, the concepts of sequential are unchanged, let $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ be a set of finite and discriminative items, $s = \langle e_1, e_2, ..., e_l \rangle$ where $e_j \subseteq \mathcal{I}, 1 \leq j \leq l$ is defined as a sequence. In addition, a sequence α is said to contain another sequence β if and only if for each element in β, there always exists an element in α that contains it without changing the order. However, the issue becomes more complicated with the introduction of the utility framework.

Generally speaking, high utility sequential patterns mining aims to discover patterns from a utility sequential database as described in Table 2.6 and Table 2.7. A big difference is that the frequent sequential pattern mining contains only one metric of frequency, while in high utility sequential pattern mining, there are several measurements to that effect.

US and UL

In 2012, Ahmed et al. introduced the concept of sequential pattern mining into the utility framework, and two algorithms were also proposed and called UL and US. These two algorithms are typical level-wise and pattern-growth algorithms as extensions of AprioriAll (Agrawal & Srikant 1995) and PrefixSpan (Pei et al. 2001). In addition, the authors also proposed a pruning strategy called SWU (Sequence-Weighted Utility), which is similar to TWU in utility mining.

2.2.4 Smart Summary

By way of a brief summary, we reorganised some of the existing work in Table 2.8 below to make a comparison of mainstream algorithms on Frequent Pattern Mining (FPM), Frequent Closed Pattern Mining (FCPM), Frequent Sequential Pattern Mining (FSPM), High Utility Pattern Mining (HUPM) and High Utility Sequential Pattern Mining (HUSPM).

Table 2.6: Quantitative Sequence Database

SID	TID	Transaction	TU
S_1	T_1	bread, milk	3
S_1	T_2	bread, milk, butter, cheese	12
S_2	T_1	bread, butter, cheese	10
S_2	T_2	milk, butter, cheese	11
S_2	T_3	bread, cheese	5
S_3	T_1	bread, milk, cheese	7
S_3	T_2	milk, cheese	6
S_3	T_3	bread, milk, butter, cheese	12

Table 2.7: Quality Table

Item	bread	milk	butter	cheese
Profit	1	2	5	4

Table 2.8: Smart Summary

Tasks / Frameworks	**FPM**	**FCPM**	**FSPM**	**HUPM**	**HUSPM**
Apriori-based (levelwise) generate-and-test	Apriori	A-Close	AprioriAll	MEU, TP, IIDS (FUM/DCG+)	UL, USpan
FP-tree-based (deepfirst) pattern-growth	FP-Growth	CLOSET	PrefixSpan	CTU-Mine/PRO/PROL, IHUP, d^2HUP	US
UP-tree-based (deepfirst) utility-pattern-growth				UP-Growth, UP-Growth+, TKU, CHUD	UP-Span
Incremental database				IHUP, FUP-HU(I), Pre-HUI	
Vertical Approaches	Eclat	CHARM	SPADE	Udepth, HUIM	

2.2.5 The Frequency-Utility Mining Model

The Association Rule Mining (ARM) is proposed to mine itemsets without knowing the profit produced. On the other hand, the High Utility Itemsets Mining (HUIM) seeks high profit items but there is no guarantee of the frequency. When supermarket managers and chainstore retailers seek for a combination of products or promotion packages which can constantly generate high profits, neither ARM nor HUIM alone can work. To this end, researchers have proposed a novel utility-frequency approach to identify all the itemsets of a user specified utility threshold and a minimum frequency threshold. Several papers with quite different approaches on this topic are proposed.

Yeh et al. proposed a bottom-up two-phase algorithm called BU-UFM (Bottom-Up Utility-Frequent Mining) for a novel utility-frequent mining model in 2007 (Yeh, Li & Chang 2007), in order to mine utility-frequent itemsets efficiently. In the paper, they propose a new definition called *Utility-Frequency (U-Frequency)*, which is neither upward nor downward and is closed with respect to the lattice of all itemsets. This is because in the new definition, the itemset which is regarded as a factor to calculate support, should be no less than a given utility threshold. Consequently, the value of new defined support is also less than traditional defined support. As the U-frequency is neither monotone nor anti-monotone, they propose a further definition called *Quasi-Utility-Frequency (QU-Frequency)* with the property of upward closed. A definition called *Qsupport* is proposed to measure whether an itemset is a QU-frequent itemset or not. In addition, Qsupport is calculated in addition to the utilities of all the items in each satisfied itemset, in which the utilities must be higher than the given utility threshold. Since a tighter upper bound is established, the QU-frequency is upward closed with respect to the lattice of all itemsets, and a Top-Down Utility-Frequent Mining (TD-UFM) is proposed as well. In 2010, Lin et al. (Lin, Zhu, Li, Geng & Shi 2010) proposed a Share (tree) strategy for Utility Frequent Pattern Mining (Share-UFPM) based on FP-tree, FP-growth and Share-tables. This

algorithm is proved to be efficient than BU-UFM.

In 2008, Khan et al. proposed a weighted utility framework for mining association rules efficiently (Khan, Muyeba & Coenen 2008). In this paper, a weighted value is given as the profit utility of an item. In addition, the Transaction Weighted Utility (TWU) is defined as dividing a positive number, which is the sum of each item's weighted utility times its quantities, by the variety number of the items. A new definition of Weighted Utility Support(WUS) is proposed as a measurement to generate association rules based on the TWUs of two items. This algorithm has proved to be faster than Weighted ARMs and standard ARM.

2.3 Actionable Combined Pattern Mining

As we discussed above, most of the research is limited to the utility-frequency cases. It also does not consider the weighted utility as the only measurement to calculate both utility and frequency and it does not regard the frequency and utility as two filters in steps, which actually makes it less representative for pattern selection. To build an actionable representativeness pattern structure is then a natural next step. However, it is not easy task.

In the section, we review an actionable combined pattern mining structure in the frequent pattern mining domain.

The concept of *Combined Association Rules* was first introduced in (Zhao, Zhang, Figueiredo, Cao & Zhang 2007), and then extended in (Zhang et al. 2008). Combined rule mining provides a new way of merging knowledge typically for scenarios such as when two features we come across to mine are not in one dataset, and merging two datasets is quite a waste of time i.e. they have different composed features as one contains customer IDs, age, address, gender, living region, nationality etc. while the other one contains gender, annual incomes, debt, the debt repayment method, and the repayment period. A preferred way is to select the features and combine them for further mining.

At the same time, high impact combined pattern mining was proposed in (Cao et al. 2007). These patterns are not necessarily frequent, but they play an important role in solving business problems by targeting those patterns of high business expectations. These patterns are exceptional because they can't be detected by traditional frequent pattern mining methods which can only find patterns with high frequency. The process to discover such actionable patterns are called AKD (Actionable Knowledge Discovery) (Cao et al. 2010). The basic framework of a combined or impacted pattern cluster is presented in Equation 2.12. For each equation, A is one feature, B is another, and C is the result or target, which might be an exceptional result if C is good enough or is too bad. Yet, this approach is only used in the frequency-based framework.

$$\begin{cases} A_1 + B_1 \rightarrow C_1 \\ A_1 + B_2 \rightarrow C_2 \\ A_1 + B_3 \rightarrow C_3 \\ ... \end{cases} \tag{2.12}$$

2.3.1 Combined Pattern Mining

Many established works on the pattern mining domain for fast and efficient mining try to tighten the upper bounds of their algorithms. It is essential to reduce the size of candidate space during each stage of the algorithm processing. One alternative choice when association rules should be mined from multiple datasets is a combined mining approach instead of a combination of multiple datasets. The reason as demonstrated is that a large combined dataset costs too much space in memory to make the algorithm efficient. In addition, it may even run out of memory (Zhao et al. 2007).

The aim of combined mining is to identify more informative knowledge that can be provided by a comprehensive presentation of problem solutions. (Cao et al. 2010). The basic framework of combined pattern mining to find combined rules from distributed datasets is as follows:

(i). The first step is to find those top-m frequent patterns from the first

dataset called initial patterns.

(ii). Based on the discovered frequent patterns, a second dataset is divided into groups, each of which is associated with a frequent pattern and a target. This step is called Segmentation.

(iii). The next step is mining the top-n frequent patterns in each group.

(iv). Find all the rules, each of which is composed of one of the top-m patterns, one of the top-n patterns and a target based on all the results discovered in the above steps. This process is called combined pattern mining to find combined rules like those described in Equation 2.12.

Compared with the conventional association rule, this new combined association rule makes it possible for users to take action directly (Zhang et al. 2008), thus it is a kind of actionable pattern (Cao et al. 2011). Combined association rules are composed of non-actionable attributes, actionable attributes, and class attributes. Combined association rule pairs and combined association clusters are built based on the same non-actionable attributes (Zhao et al. 2008, Cao et al. 2011). For example, in Equation 2.12, A_1 is the non-actionable attribute, and same in all three equations. B_1, B_2 and B_3 (B_i is applied to describe each element in this cluster)are three actionable attributes. Furthermore, different actions lead to different results e.g. C_i, which is a class attribute. Thus, for a group of objects having the same non-actionable attributes A, the actions B_i correspond to a preferred class C_i that can be performed directly.

Zhang et al. (Zhang et al. 2008) proposed a new lift definition called *conditional lift* to measure the interestingness of a combined association rule as follows.

$$ConLift = \frac{Conf(A + B_i \Rightarrow C_j)}{Conf(A \Rightarrow C_j)} = \frac{Count(A \cap B_i \cap C_j) * Count(A)}{Count(A \cap B_i) * Count(A \cap C_j)} \tag{2.13}$$

Where *ConLift* means the conditional lift of combined association rule demonstrated in Equation 2.12. *Count* is the count of the appearance time of the itemsets. In addition, $A \cap B_i \cap C_j$ stands for the situation that A, B_i and C_j occur simultaneously.

Later, Zhao et al. (Zhao et al. 2008) give a specific definition on combined association rules, which is:

Definition 2.10 *Assume that there are k datasets D_i $(i = 1...k)$ and I_i to be the set of all items in datasets D_i. $\forall i \neq j$, $I_i \cap I_j = \emptyset$. A combined association rule can be represented as*

$$A_1 \wedge A_2 \wedge ... \wedge A_k \rightarrow T \tag{2.14}$$

where $A_i \subseteq I_i (i = 1...k)$ is an itemset in dataset D_i, $T \neq \emptyset$ is a target item or class and $\exists i, j, i \neq j, A_i \neq \emptyset, A_j \neq \emptyset$.

For example, A_1 is a demographic itemset, A_2 is a transactional itemset in a marketing campaign, A_3 is an itemset from a third-party dataset and T is the loyalty level of one customer.

Further more, Zhao has given the definition of the combined rule pair and combined rule cluster. A combined rule pair, as shown in Equation 2.15, is a pair of combined association rules containing one of the same characteristics U and different policies or actions V_1 and V_2 which lead to different targets or results T_1 and T_2. In addition, a combined rule cluster is a cluster of combined association rules as shown in Equation 2.16. In Cao's journal article (Cao et al. 2011), the rules to be discovered using the equation 2.16 can be based on one single dataset after dividing the dataset into three parts: non-actionable part U with multiple non-actionable attributes, actionable part V_i with multiple actionable attributes, and class part T with one class attribute. Mining these combined rules can be applied in two phases: 1) Mining the frequent non-actionable itemsets ID, and finding the frequent itemsets including the class part I_{DC}; 2) Discovering the patterns including three parts of the itemsets I_{DCA}, comparing it with the minimum threshold

and adding it to the combined patterns set. This framework can also be applied to imbalanced data mining to avoid some risks in market investment and many other domains.

$$P : \begin{cases} R_1 : U + V_1 \rightarrow T_1 \\ R_2 : U + V_2 \rightarrow T_2 \end{cases} \tag{2.15}$$

$$C : \begin{cases} U + V_1 \rightarrow T_1 \\ U + V_2 \rightarrow T_2 \\ ... \\ U + V_n \rightarrow T_n \end{cases} \tag{2.16}$$

In addition, two new lifts are defined to measure the interestingness of combined association rules.

$$Lift_U(U \wedge V \rightarrow T) = \frac{Conf(U \wedge V \rightarrow T)}{Conf(V \rightarrow T)} = \frac{Lift(U \wedge V \rightarrow T)}{Lift(V \rightarrow T)} \tag{2.17}$$

$$Lift_V(U \wedge V \rightarrow T) = \frac{Conf(U \wedge V \rightarrow T)}{Conf(U \rightarrow T)} = \frac{Lift(U \wedge V \rightarrow T)}{Lift(U \rightarrow T)} \tag{2.18}$$

$Lift_U(U \wedge V \rightarrow T)$ is defined to describe how much U contributes to the rule and can be regarded as the lift of U with V as a precondition. The interestingness of combined association rules is defined as Equation 2.19 based on the two new lifts.

$$I_{rule}(U \wedge V \rightarrow T) = \frac{Lift_U(U \wedge V \rightarrow T)}{Lift(U \rightarrow T)} = \frac{Lift_V(U \wedge V \rightarrow T)}{Lift(V \rightarrow T)} \tag{2.19}$$

In 2010, Cao et al. further consolidated the existing works and came up with the concept of *Combined Mining*, which was newly defined and regarded as one of the general methods for directly discovering informative and decision-making patterns via analysis of complex data from multiple sources or with heterogeneous features and attributes to satisfy business and industrial expectations instead of academic purposes. In conclusion, the general ideas for combined mining are as follows.

- To reflect multiple aspects of concerns and characteristics in business and industry by involving multiple heterogeneous features.
- To reflect multiple aspects of nature recorded across the business lines by discovering multiple data sources.
- To disclose a deep and comprehensive understanding of the data by applying multiple methods.
- To reflect concerns and significance from multiple perspectives with the application of multiple interestingness metrics in technology and business.

Cao had spent more effort on summarising and abstracting several general and flexible frameworks (Cao et al. 2010, Cao et al. 2011, Cao 2012) rather than presenting several specific algorithm for mining particular types of combined patterns for the reasons below:

- The general and flexible frameworks from an architectural perspective can foster wider implications and be instantiated into specific combination methods and algorithms for specific purposes.
- In general, the generalisation capability of the proposed framework determines its value. Presenting the basic concepts, paradigms and processes is another way to show the valuable combined mining framework.
- It is very important to prove the generalisation capability for producing proposed combined patterns, which can be generated by different methods, from different data sources and different features.
- In addition, Taking the relationship analysis into consideration in constructing the framework is also very important because the relationship among the atomic patterns within a combined pattern affects how the combination is generated and measured.

There are many kinds of pattern relations (Cao 2012), including but not limited to the relation mentioned below:

- Serial Coupling, each pattern is in time serial order.
- Causal Coupling, each pattern is in relation to an exact causality.
- Synchronous Coupling, each pattern occurs at the same time.
- Conjunction Coupling, each pattern always appears together.
- Disjunction Coupling, at least one pattern will happen.
- Exclusive Coupling, at most one pattern will happen.
- Dependent Coupling, some of the patterns hold a dependent relation and always happen together.

In addition, just take the serial coupling relation into consideration, there might be some sub-relations between patterns:

Positive enabling relation,	e.g. $a \rightarrow b$
Negative enabling relation,	e.g. $a \rightarrow \neg b$
And split relation,	e.g. $a \rightarrow (b \wedge c)$
Or split relation,	e.g. $a \rightarrow (b \vee c)$
And join relation,	e.g. $(a \wedge b) \rightarrow c$
Or join relation,	e.g. $(a \vee b) \rightarrow c$

2.3.2 Actionable Knowledge Discovery

There is no doubt that Cao's group are the pioneers of actionable knowledge discovery. They first realised that there exists a big gap between academic results and business or industrial expectations in 2007 (Cao et al. 2007). For example, in the real world, exceptional behaviour can be seen in many situations. Such behaviour is rare and dispersed, while some of it may be

associated with a great impact or a significant or even disastrous effect on society. In addition, the exceptional behaviour data is organised into four datasets: the unbalanced set, balanced set, target set, and non-target set. Because of the unbalanced class distribution and unbalanced itemset distribution, a balanced activity set is necessary to extract the same number of non-target activity as that of the target activity. In this way, impact-targeted exceptional behaviour patterns easily stand out from overwhelming non-impact itemsets. In addition, two kinds of support are defined to reflect either the global statistical significance (in Equation 2.20) or the impact-oriented statistical significance (in Equation 2.21) of one behaviour based on a pattern $\{ P \rightarrow T\}$

$$Supp_A(P,T) = \frac{|P,A|}{|A|} \tag{2.20}$$

$$\begin{cases} Supp_D(P,T) = \frac{|P,D|}{|D|} \\ Supp_{\overline{D}}(P,\overline{T}) = \frac{|P,\overline{D}|}{|\overline{D}|} \end{cases} \tag{2.21}$$

Cao noticed types of high impact exceptional behavior patterns which had rarely been studied by academics. These are listed below.

- Positive Impact-Oriented Exceptional Behaviour Patterns. In this condition, the ratio that $Supp_D(P,T)/Supp_{\overline{D}}(P,\overline{T})$ should be larger than a given threshold. This ratio indicates the difference between the target and non-target set.

- Negative Impact-Oriented Exceptional Behaviour Patterns. In this condition, the ratio that $Supp_A(P,T)/Supp_A(P,\overline{T})$ should be larger than a given threshold. This ratio indicates the statistical difference of a pattern P leading to a positive or negative impact in a global manner.

- Impact-Contrasted Exceptional Behaviour Patterns. In this condition, one of the following two scenarios must be satisfied.

  $Supp_D(P,T)$ is high but $Supp_{\overline{D}}(P,\overline{T})$ is low.

$Supp_D(P, T)$ is low but $Supp_{\overline{D}}(P, \overline{T})$ is high.

The big contrast between two supports indicates that P is more or less associated with a positive rather than negative impact, or vice versa.

2.4 Summary

In this section, we present a brief summary of the literature review and the foundation part.

In section 1, the frequency/support framework is discussed. We first introduce the frequent mining framework with basic definitions and concepts. For example, the definition of itemset and the support and the Downward Closure Property(DCP). Then we talk about the application and association rules mining in terms of the definition of confidence. In addition, some fast algorithms are presented for frequent pattern mining, closed and maximal frequent itemset mining, top-k frequent itemset mining and frequent sequential pattern mining. Finally, we discuss the weighted frequent itemset mining as a branch in the frequent pattern mining domain.

In section 2, the utility framework is discussed. Also, we present a brief introduction on the utility framework. A dataset contains two kinds of utilities: a profit utility and a quantity utility. The utility of an item in a given transaction is the product of these two values in the transaction. We search for the papers on utility mining as much as possible, and split them into two parts, high utility itemset mining and high utility sequential pattern mining. For different parts, we have Apriori based, FP-tree based, UP-tree based and vertical approaches. We also list one major application as for mining from the incremental database. Lastly, we discuss some frequency-utility mining model.

As the existing frequency-utility models are not suitable for mining both high utility and high frequency patterns for representativeness as well as actionable directed knowledge, we introduce the concept of actionable combined mining for application in the frequent mining domain. Further, we try

Table 2.9: Location of Proposed Approaches

	Frequency/Support Framework	Utility Framework
Academic Approach	Frequent Pattern Mining	High Utility Itemsets Mining
Actionable Approach	Combined Pattern Mining	**Our Approaches**

to capture its main idea and maintain its outstanding performance in the utility domain.

Our work in this thesis is located in Table 2.9.

Chapter 3

Mining Combined High Utility and Frequent Patterns

3.1 Introduction and Background

While the utility-based framework greatly enhances the actionability (Cao et al. 2010) of resultant patterns, compared to frequent pattern mining, it is still defective in some circumstances. As in utility mining, the unit profit of each item is given, while the quantity of an item depends on the transaction. Mining the utility of an itemset can be regarded as a statistical way to discover the itemsets whose utility is larger than a specific value (Tseng et al. 2010). However, the use of utility alone makes it ineffective to discover strongly associated items. For instance, selling a *pedigree cat* in a pet store happens maybe once a month or even more rarely. Although the profit is extremely high, it is probably not very wise for a manager to spend too much on designing strategies to promote such kind of the pedigree pet with pet foods, because selling the pedigree pet could just be a coincidence. Furthermore, if a customer happens to purchase many other items at the same time, patterns like "pedigree cat, cat food, cattery, troughs" or "pedigree cat, collar, cattery, cat litter" etc. could be selected as high utility itemsets. Obviously, such itemsets are neither representative nor actionable to the manager.

Table 3.1: An Example Database

TID	Transaction	TU
T_1	(A, 1) (C, 1) (D, 1)	8
T_2	(A, 1) (B, 6) (C, 2) (F, 5)	24
T_3	(A, 2) (B, 2) (C, 6) (D, 5) (E, 1)	60
T_4	(B, 4) (C, 3) (D, 2) (F, 3)	18
T_5	(B, 2) (C, 2) (F, 3)	9

Table 3.2: Profit Table as an Example

Item	A	B	C	D	E	F
Profit	5	2	1	2	30	1

Such situations are described in Table 3.1 and Table 3.2 as an example. All subsets belong to T_3, containing {F} are finally proved to be HUIs, while the given threshold is no less than 50. All other itemsets are filtered as they fail to pass the threshold. Obviously, such itemsets are neither representative nor actionable to most of businessmen. However, if the threshold is set too low, new problems will appear. Searching a large number of itemsets on a large dataset may encounter a large search space.

With the examples in the above tables, Table 3.3 lists the utility, support and confidence of itemsets (here each rule is also called an itemset). The support of an itemset reduces when the itemset size (length) increases because of the *Downward Closure Property (DCP)*(Agrawal et al. 1994). However, some patterns still contain more information than others. For example, the association rule $\{C \rightarrow B\}$ is not of a high confidence, but is associated with a

Table 3.3: A Comparison of Itemset Utility, Support and Confidence

Itemset(Pattern)	Utility	Support	Confidence
{A}	20	60%	Nil
{A → E}	40	20%	33%
{C}	14	100%	Nil
{C → B}	41	80%	80%
{BC → E}	40	20%	25%

high utility increment from {C} (with utility 14) to {BC} (41), which should be more interesting than other rules in this table.

To overcome the issues and challenges discussed in Chapter 1, a new framework is required to discover the really actionable patterns: they are not only succinct in terms of presentation (for a given item, only the most profitable itemset instead of many should be chosen) but also actionable (both utility-contrasted and frequent). Even though this seems to be very promising and interesting to users, it is critically challenging to build such a framework. In addition, Fig. 3.1 illustrates the utility dynamics of utility-based itemsets when itemset length grows.

Since there are significant gaps existing in objectives and definitions between utility-based itemsets and association rules, it is hard to simply merge them. We introduce the concept of *combined mining* (Cao et al. 2007) to combine the utility-based framework and basket analysis. In combined mining, taking Fig. 3.1 and Table 3.4 as an example, *Derivative Itemset (DI) (e.g. X_a, X_b)*, also called combined itemset, is an itemset consisting of two parts. One part is called *Underlying Itemset (UI)*, which is the same part X_0 shared in both X_a and X_b in Equation 3.1. The other part is called

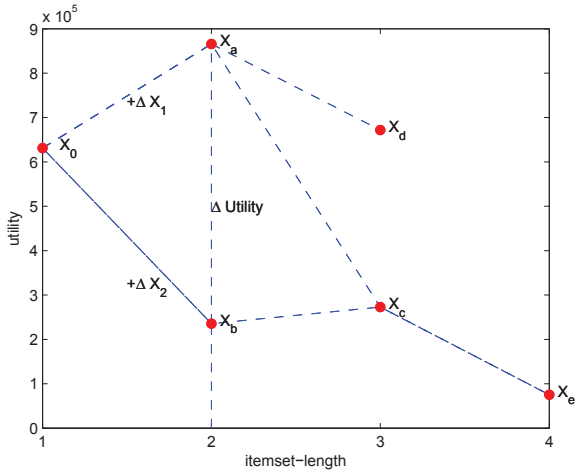


Figure 3.1: Example of utility dynamics in terms of itemset growth

Table 3.4: Pattern Expression

Utility-Association	Traditional Association	Structure of Pattern Relation
$X_0 \rightarrow X_a$	$X_0 \rightarrow \Delta X_1$	$X_0 + \Delta X_1 = X_a$
$X_0 \rightarrow X_b$	$X_0 \rightarrow \Delta X_2$	$X_0 + \Delta X_2 = X_b$

Additional Itemset (AI), which is different in (a) and (b), marked as ΔX_1 and ΔX_2. The combined patterns are called 'Utility-Association Combined Patterns' as shown in Table 3.4.

$$\begin{cases} X_0 \rightarrow X_a & (a) \\ X_0 \rightarrow X_b & (b) \end{cases} \tag{3.1}$$

With a single underlying itemset, a cluster of *DI*s might be discovered with respect to different *AI*s. Here we define actionable high utility itemsets as a pair of *DI*s. Furthermore, we note that one of the pairs has the highest utility among all the *DI*s with the same *UI*, whereas the other *DI*s have the lowest utility. This type of combined high utility pattern is informative for decision-making. For instance, in marketing, it may suggest a manager that

some products should be sold with the others for high profit, whereas the same products sold with something else may result in a loss. Obviously, such kinds of combined patterns incorporate item and itemset relationship and utility, and are thus more actionable for decision making.

In this chapter, we propose a novel pattern structure called *Actionable Combined Utility-Association Rule (CUAR)* and a new interestingness coefficient for selecting patterns called *Associated-Utility Growth (AUG)* to help us discover such patterns with both high utility and high frequency.

The rest of this chapter is organised as follows. In Section 2, the background is introduced. The problem is stated in Section 3. In Section 4, we propose the *CUARM* algorithm. Section 5 presents the experimental results, and Section 6 gives a brief summary of this work.

3.2 Problem Statement

In this section, we define *Associated-Utility Growth (AUG) Pattern*, which introduces both association and utility growth into combined mining and mining actionable combined pattern pairs.

3.2.1 Preliminaries

Taking Table 3.1 and Table 3.2 as an example, let $D = \{T_1, T_2, ..., T_m\}$ be a transaction database, and $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ be a set of finite and discriminative items. Each transaction $T_c \in D$ $(1 < c < m)$ is a subset of $\mathcal{I}$ with a distinct identifier called *TID*. In a given transaction T_c, each item i_k appearing with a positive integer, $q(i_k, T_c)$ is called i_k's *quantity utility* in T_c. Also, each item in $\mathcal{I}$ is associated with a positive number $p(i_k, \mathcal{I})$, which is called i_k's *profit utility* in $\mathcal{I}$.

Definition 3.1 *The* frequency *of an itemset X counts the times it appears in all transactions and is denoted as $SC(X)$. The* support *of X is $SC(X)$, divided by the number of transactions in D, and is denoted as $supp(X)$.*

Based on Table 3.1, $SC(A)$ is 3, $SC(AB)$ is 2, and $Supp(AB) = 40\%$.

Definition 3.2 *The* utility of item i_k *in a transaction* T_c *is the profit utility of the item times its quantity utility in a transaction, defined as*

$$u(i_k, T_c) = p(i_k, \mathcal{I}) * q(i_k, T_c) \tag{3.2}$$

An item i_k with its utility in a transaction T_c is denoted as $u(i_k, T_c)$ $(i_k \in T_c)$.

Definition 3.3 *The* utility of an itemset X *in a transaction* T_c *is the utility sum of all items belong to the itemset, defined as*

$$u(X, T_c) = \sum_{i_k \in X} u(i_k, T_c) \tag{3.3}$$

An itemset containing l discriminative items is called an *l-length* itemset, where $X \subseteq \mathcal{I}$. The utility of the same item in different transactions might be different considering the quantity of each item purchased.

Definition 3.4 *The* utility of an itemset X *in the whole database* D *is the sum of the utility of this itemset in all transactions. It is denoted as* $U(X)$*, and defined as*

$$U(X) = \sum_{X \subseteq T_c \wedge T_c \in D} u(X, T_c) \tag{3.4}$$

In addition, one item can be regarded as an 1-length itemset.

Definition 3.5 *The minimum utility threshold is denoted as* min_util*, and a set of all itemsets whose utilities are higher than* min_util *is denoted as* $f_H(D, min\_util)$*. The goal of HUI mining is to find such an itemset,* $f_H(D, min\_util)$*.*

Definition 3.6 *The* transaction utility *of the transaction* T_c *is denoted as* $TU(T_c)$ *and defined as*

$$TU(T_c) = \sum_{i_k \in T_c} u(i_k, T_c) \tag{3.5}$$

Definition 3.7 *The* transaction-weighted utility *of the itemset X is the sum of the transaction utilities of all the transactions that X belongs to. It is denoted as $TWU(X)$ and defined as*

$$TWU(X) = \sum_{X \subseteq T_c \wedge T_c \in D} TU(T_c) \tag{3.6}$$

Definition 3.8 *The* high transaction-weighted utility itemset (HTWUI) *consists of those itemsets whose TWU is no less than* min_util.

Property 3.1 *The* transaction-weighted downward closure property *holds for HUI, says that if an itemset X is not a HTWUI, all its supersets are not HUIs because $U(X) \leq TWU(X)$.*

3.2.2 Mining High Combined Utility-Association Rules

HUI Mining discovers itemsets with high utility whose utilities are higher than the minimum threshold, in which their frequencies are not concerned. HUI could be regarded as an extension of FIM towards addressing business interest (Cao et al. 2010) represented by utility. Large number of itemsets will come out if the threshold is not right. As shown in Fig. 3.1 of the utilities in a cluster of incremental itemsets, we can see the utility is changing dynamically and irregularly.

As the basic rule in social marketing is to gain profit, businessmen might only care about products that can make profit for them, and are also interested in converting those less popular goods to those that are preferred.

It is helpful for business purposes to figure out those itemsets which 1) are low utility itemsets, but whose utility becomes high after one additional item (or itemset) is added; 2) are high utility, but become low utility after one item (or itemset) is added. An itemset whose length increases with adding new items is called *Incremental Itemset*, which means the number of items it contains grows but never reduces.

Even though the utility metric provides reasonable evidence for selecting patterns of business interest, it does not provide sound insurance about how

sound a high utility pattern can be, when complemented by frequency-based filters. Therefore, we propose a new strategy which combines both utility and frequency, named as *Associated-Utility Growth (AUG) Pattern Mining*, to identify *Association-Maximum Incremental Itemsets (AMII)* and *Utility-Increasing Incremental Itemsets (UIII)*. The utility of an incremental itemset is dynamic, meaning that the utility evolves when additional items are added to the underlying items to form a utility curve. A UIII structure is necessary to discover those utility increment-oriented itemsets. Also, there is no doubt that the frequency of an incremental itemset monotonically decreases. Here maximum association does not refer to those itemsets with the highest frequency, but those itemsets where items share a reasonable relationship with each other. In some way, it also means that the frequency would not change too much after adding one or more items. Subsequently, the measurement of AUG is considered for candidate pruning to find out those having both high utility growth and highly associated items. In this way, only one significant combined pattern is selected for each UI.

3.2.3 An Abstract Model: 2-length Combined Utility-Association Pattern Pair

Here we illustrate the application of *actionable Combined Utility-Association Rules* through identifying *2-length Combined Utility-Association Pattern Pair*. Take Fig. 3.1 as an example (here we suppose they hold the minimum confidence threshold, or X_a and X_b share the same relation with X_0), the 1-length itemset X_0 is firstly treated as a *UI*, then two items added separately form two 2-length itemsets: ΔX_1 is added and forms one new itemset X_a with higher utility, ΔX_2 is added and forms the other new itemset X_b with lower utility. X_a and X_b are two supersets of X_0. The pattern pair is shown in Equation 3.7.

$$\begin{cases} X_0 \rightarrow X_a \Rightarrow U - Increase & (a) \\ X_0 \rightarrow X_b \Rightarrow U - Decrease & (b) \end{cases} \tag{3.7}$$

Definition 3.9 Positive Impact Rule (PIR)*: referring to rules structured as Equation 3.7(a) which is called positive impact rules, whose right-hand side is associated with utility higher than X_0 on the left-hand side.*

Definition 3.10 Negative Impact Rule (NIR)*: referring to rules structured as Equation 3.7(b) which are called negative impact rules, those on the right-hand side are associated with a utility which is lower than X_0 on the left-hand side.*

If such rules are used for marketing purposes, a retailer should know what promotion mixtures make more profit and what leads to low profit if they are put together, based on the positive and negative rules.

3.3 The CUARM Approaches

We aim to provide patterns to retailers for promotion strategies including increasing high utility product combinations which are highly associated with each other. We name this the *Utility-Association Rules*, which cannot be discovered by traditional association rule methods or utility mining algorithms alone. The algorithm for identifying interesting utility-association rules is called *Combined Utility-Association Rules Mining (CUARM)*. To this end, two factors, *Contribution* and *Weight*, are proposed to select combined patterns of both high utility growth and strong association.

3.3.1 The Baseline Approach

As stated above, the purpose of HUI mining is to find the set of all high utility itemsets $f_H(D, min\_util)$ efficiently. One way to obtain the target patterns is to obtain $f_H(D, 0)$ first, which can be achieved by using UP-Growth with $min\_util = 0$, and then extracting the UARs from it. In essence, the baseline approach is a strategy to maintain all itemsets with their utilities. Readers can refer to (Tseng et al. 2010) for detailed structure and examples about

Table 3.5: Reorganized Transactions with Their Reorganized-TUs

TID	Transaction	RTU
T'_1	(C, 1) (A, 1) (D, 1)	8
T'_2	(C, 2) (B, 6) (A, 1) (F, 5)	24
T'_3	(C, 6) (B, 2) (A, 2) (D, 5) (E, 1)	60
T'_4	(C, 3) (B, 4) (D, 2) (F, 3)	18
T'_5	(C, 2) (B, 2) (F, 3)	9

Table 3.6: Items with Their TWUs

Item	A	B	C	D	E	F
Profit	92	111	119	77	60	51

UP-Growth and UP-Tree. In addition, the UG-Tree we propose is built with $min\_util = 0$ based on Table 3.5 and Table 3.6.

3.3.2 The Proposed Approach

By using the UP-Tree, which contains enough information, we can generate the utility-association rules. While the UP-Tree is retrieved, the utility of each itemset can be discovered and prepared for the calculation of *Factor C*. At the same time, the support count of each item can also be found through the tree and used for *Factor W*. These two factors will be discussed in the next subsections.

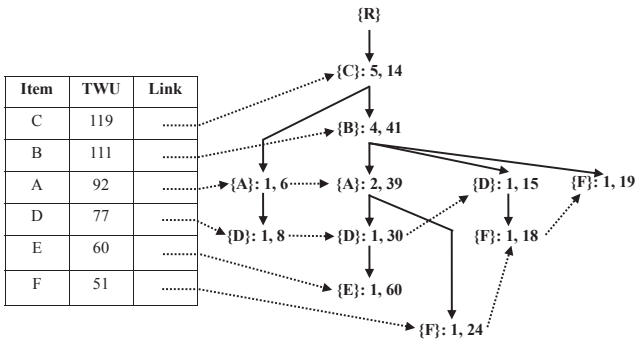


Figure 3.2: Header table and a UP-Tree when *min_util = 0*

3.3.3 Impact Factor of Utility Growth across Combined Itemsets

Both PIRs and NIRs are not difficult to be acquired, since just scanning the database one more time with a comparison added will help. However, this belongs to the post-processing approach which is less efficient, and ignores item relation analysis during the mining process. For these, a measurement called *contribution* is proposed below to discuss the relationship among these three itemsets (UI, AI and DI), on top of PIRs and NIRs.

Definition 3.11 *The* contribution *of Additional Itemset (ΔX) to make utility change (increase or decrease) from the Underlying Itemset (X_0) to Derivative Itemset (X), denoted as $C(\Delta X|X_0)$, is defined as:*

$$C(\Delta X|X_0) = \begin{cases} \frac{2}{1+e^{-\mathcal{R}}} - 1 & , \quad U(X) > U(X_0) \\ \mathcal{R} & , \quad U(X) \leq U(X_0) \end{cases} \tag{3.8}$$

In Equation (3.8),

$$\mathcal{R} = \frac{U(X)}{U(X_0)} \tag{3.9}$$

This equation is proposed to measure whether *itemset* ΔX, as an additional part of X_0, plays an important role in transforming itemset X_0 to X in

the utility perspective. Here, $U(X)$ is the utility of X in the whole database. The first equation is associated with utility increase, corresponding to PIR, while the second corresponds to NIR. In the first function, a logistic function is used to converge the contribution to the range of [0,1], which will be discussed later in the next section.

As demonstrated in Fig. 3.1, the function of utility and item-length is neither monotonic nor anti-monotonic, meaning the utility of an itemset is dynamic with its length increasing. The utility of each item is taken into consideration to analyse the contribution. Furthermore, the contribution of additional itemset provides influence within the itemset sharing the same support counts, which is not suitable for the operation of contribution. However, even though the contribution is presented to measure whether ΔX plays a significant role to promote the utility from X_0 to X, it is still measured by the rate $\mathcal{R}$ because ΔX might appear in other itemsets, and the utility should be calculated in another way.

Two conditions might appear: 1) The contribution of this ΔX is high enough to make $X(DI)$ a significant PIR, which also means itemset X_0 has a strong utility-association with itemset X; 2) The contribution of this ΔX is so low that this $\{X_0 \rightarrow X\}$ is proved to be a significant NIR. As a result, itemset X_0 is rarely utility-associated with itemset X.

3.3.4 Co-occurred Associations between Underlying and Additional Itemsets

Definition 3.12 *The weight of an additional itemset to measure the co-occurrence frequency of the underlying itemset and additional itemset, denoted as $W(\Delta X|X_0)$, is defined as:*

$$W(\Delta X|X_0) = \frac{Supp(X)}{Supp(X_0 \cup \Delta X)} \tag{3.10}$$

It is a reduction of the Jaccard similarity coefficient [1]:

$$J(X_0, \Delta X) = \frac{|X_0 \cap \Delta X|}{|X_0 \cup \Delta X|} \tag{3.11}$$

This equation aims to examine whether itemset ΔX has a high or low association by measuring their co-occurring frequency with itemset X_0.

In Equation 3.10, *Supp(X)* is the support of X_0 and ΔX appearing together, and *Supp($X_0 \cup \Delta X$)* is the support of either X_0 or ΔX appearing:

$$Supp(X_0 \cup \Delta X) = Supp(X_0) + Supp(\Delta X) - Supp(X) \tag{3.12}$$

3.3.5 Impacted Coefficient of the Additional Itemset

Definition 3.13 *The impacted coefficient of an additional itemset describes how effective this itemset is to manufacture the derivative itemset from the underlying itemset, denoted as $AUG(\Delta X|X_0)$, defined as:*

$$AUG(\Delta X|X_0) = \sqrt{\frac{C^2(\Delta X|X_0) + W^2(\Delta X|X_0)}{2}} \tag{3.13}$$

This equation averages the value of $C(\Delta X)$ in Equation 3.8 and $W(\Delta X)$ in Equation 3.10. Here we use the *Quadratic Mean (QM) (also known as Root-Mean Square)* to measure the significance of the itemset ΔX in terms of both utility and relationship perspectives because it represents the sample standard deviation of the difference between W and C, thus the result cannot be affected heavily by the smaller value. It is easy to prove:

$$QM^2(X) = (\overline{X})^2 + \sigma^2(X) \tag{3.14}$$

Here, $\overline{X}$ and $\sigma(X)$ stand for the arithmetic mean and the standard deviation of W and C. We have also tried another measurement by *Harmonic Mean (HM)* as a baseline, which has proven to be less effective in our experiments.

[1] `http://en.wikipedia.org/wiki/Jaccard_index`

For a specific X_0, for each itemset ΔX to be considered, the higher AUG means this itemset is likely to impel the underlying itemset into higher utility itemset. On the contrary, the lower the AUG is, the lower utility that derivative itemset might be. As all the AUG would be calculated, only the largest AUG value itemset will be chosen.

3.3.6 The CUARM Algorithm

In this section, an algorithm named *Combined Utility-Association Rule Mining (CUARM)* is proposed to discover all the actionable combined utility-association rules. At the beginning of the algorithm, it picks all UIs as candidates. For each UI, all the combined patterns are discovered with their AUGs which form a combined pattern cluster, and only the most effective pattern is selected. In addition, if two patterns are coupled with utility increment and decrement, a combined pattern pair forms.

The input is the transaction database, including all transactions with the utility of each item, and the output is the combined pattern pairs, their underlying itemset and the corresponding utilities. In line 1, we prepare all the itemsets with their utilities in alphabetical order and the length of longest itemset. In lines 2-5, we start with each of the UIs named $itemset_0$ with its utility U_0. In lines 6-11, the DIs are ready and we calculate their AUGs. In line 12-13, we select the pattern with max AUG values as CUAR.

3.4 Experiments

In this section, we conduct intensive experiments to evaluate the proposed methods. Our experiments are run on a PC with a 2.30 GHz Intel Core, 16 gigabyte memory. CUARM is implemented in Java. Two real datasets and two synthetic datasets are used for the experiments. The real datasets are *Retail*[2] and *Chainstore*[3], and the synthetic datasets are *t20i6d100k* and

[2] `http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php`
[3] `http://cucis.ece.northwestern.edu/projects/DMS/MineBench.html`

Algorithm 3.1 CUARM

Input: Transaction database D, including the utility $U(X)$ of each item in D

Output: All actionable combined utility-association rules

```
1 Get all itemsets' utilities via UP-Tree  Get the length of longest itemset: lmax  for len = 1, len < lmax, len++ do
2     for Itemset whose length is equal to len do
3         Get itemset_0 with U_0(itemset-utility)  for itemset.length > len do
4             Check inclusive and utility changes  Get itemset_1 with U_1  Calculate C  Scan the database, get W  Calculate AUG
5         Selected max one  Present this utility-association rule
```

c20d10k. The parameters of the datasets are listed in Table 3.7.

3.4.1 Comparison of Two Functions for Calculating Impacted Coefficient

Here we propose two functions for calculating the impacted coefficient. One is the quadratic mean (QM), which is adopted in this paper, the other function is the harmonic mean (HM), which has proved to be less accurate in the experiments. Those itemsets with a good coefficient measurement are associated with both high frequency and high utility growth, we can thereby separate the database randomly. If the output itemsets discovered in each sub-database are stable, we can assume that this measurement is suitable.

The experiments are conducted on the Retail dataset for the sake of simply examining the QM function. The top 100 experimental results are selected and shown in Fig. 3.3. The figure on the left shows the comparison between UP-Growth and QM, while the figure on the right shows the result of QM and HM on $C(\Delta X)$ and $W(\Delta X)$. The database is split into 10 parts randomly. The first part contains 10% transactions in the database and each later part contains 10% more transactions than the former part (such that

Table 3.7: Characteristics of Datasets

Dataset	Number of Transactions	Number of Items	Average Length
Retail[4]2	88162	16470	10.3
Chainstore[5]3	1112949	46086	7.3
t20i6d100k	100000	658	13.7
c20d10k	10000	187	13

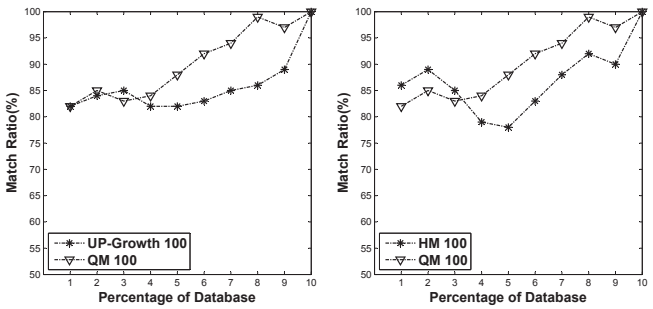


Figure 3.3: Comparison of HM, QM and UP-Growth

the second part contains 20% and the last part is 100%). The X axis is the k_{th} ($1 \leq k \leq 10$) part of the database, and the Y axis is the match ratio, which means the ratio of the exact patterns found in the k_{th} part matching with the $(k+1)_{th}$ part. As seen from the figures, the QM method outperforms both HM and UP-Growth.

3.4.2 Experimental Evaluation of CUARM

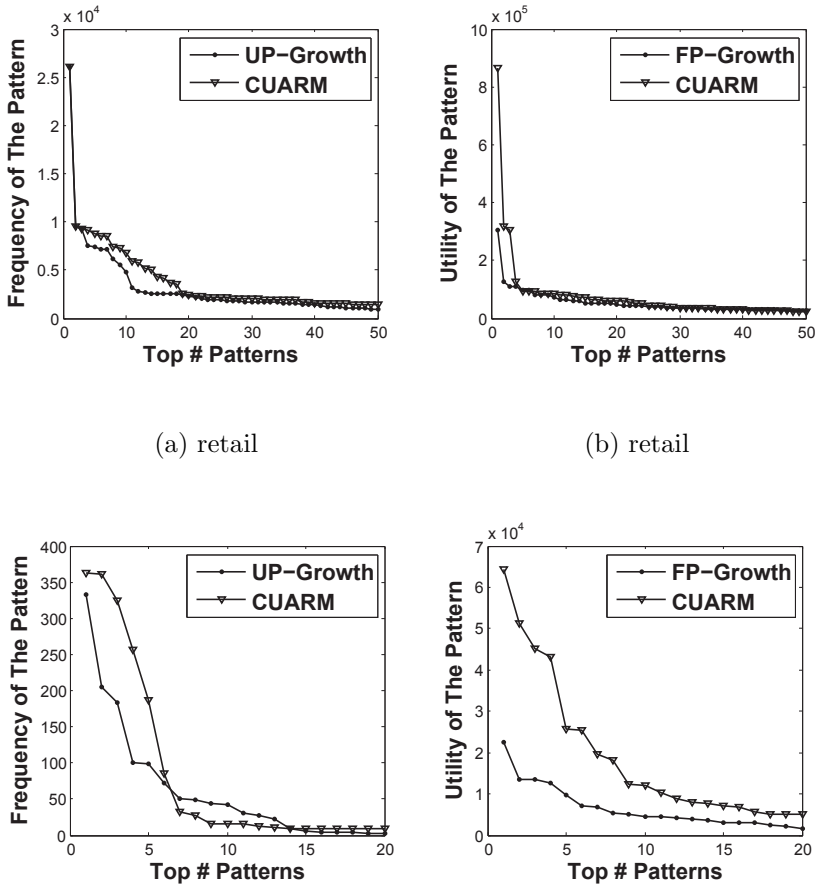
Next, we present the experimental results of comparing DIs (Derivative Itemsets) with the traditional HUIs, FIs and UIs (Underlying Itemsets) respectively. The statistic values of each dataset are shown in Table 3.7. The experiment is conducted as follows. Firstly, we collect all the utility itemsets with their utilities and frequencies in each dataset respectively. Secondly, we also collect all the frequent itemsets with their frequency and utility. Then we calculate the utilities of the FIs, frequencies of the HUIs and both utilities and frequencies of the DIs. At last, we plot the frequencies and utilities of the DIs, HUIs, FIs and UIs as shown in Fig. 3.4, Fig. 3.5 and Fig. 3.6. Such exhibition is made for the purposes of comparing our algorithm with FIM and HUI to demonstrate the Utility-Association Rules we have discovered both at high utility and high frequency.

Experiments on Real Datasets

We first present the outputs of dataset *Retail* in Fig. 3.4(a) and Fig. 3.4(b). Top 50 patterns of each algorithm are selected for experiments. By analysing the frequencies and utilities of patterns, many of them are without much difference in both experiments, which means the association rules we found via traditional AR algorithms are also high utility-association rules via our method. In addition, such rules are also associated with high utilities. This explains why some parts of the curves overlap. In addition, customers prefer to buy a few products at one time, that is, most of FIs and HUIs contain only one or two items, which also explains the observations.

In datasets *Chainstore*, the differences are much clearer, because customers usually prefer a variety of products in each of transactions, and the

frequency of high utility itemsets composed of relatively more items are always low, while highly frequent itemsets with relatively less items have low utilities. For example, in Fig. 3.4(c), the CPUIM performs much better than that of FP-Growth, while in Fig. 3.4(d), even at some points, the performance is not so good and the global performance is much better. To sum up, we can assert that the performance of our algorithm CPUIM is much better than the others.



(a) retail (b) retail

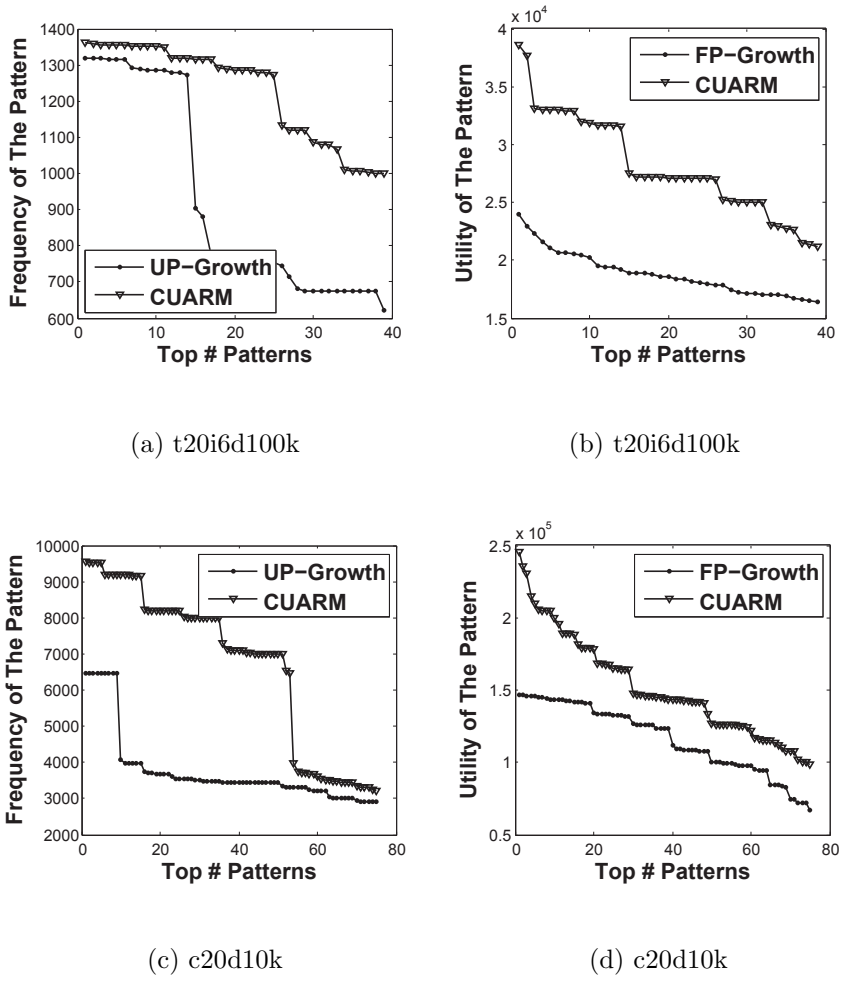
(c) chainstore (d) chainstore

Figure 3.4: Experiments for FP, UP and CUARM on real datasets

Experiments on Synthetic Datasets

Experimental results on synthetic datasets *t20i6d100k* and *c20d10k* are

shown in Fig. 3.5(a), Fig. 3.5(b), Fig. 3.5(c) and Fig. 3.5(d). The results are much clearer than those from real datasets because the items included are much neater and more orderly. For most of the patterns discovered via CUARM, the frequencies are much higher than traditional high utility itemsets. At the same time, most Utility-associated rules are also much higher, i.e., twice the utility, than traditional association rules, especially in Fig. 3.5(b) from dataset *t20i6d100k*.



(a) t20i6d100k

(b) t20i6d100k

(c) c20d10k

(d) c20d10k

Figure 3.5: Experiments for FP, UP and CUARM on synthetic datasets

3.4.3 Evaluation of the Utility Increment

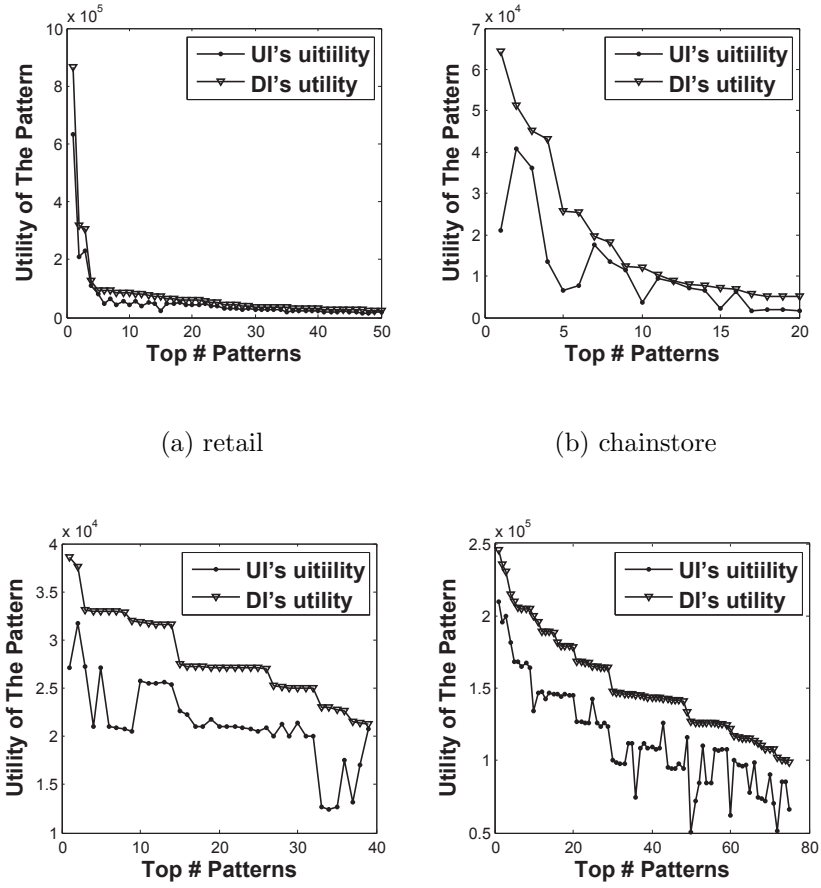
Here, we demonstrate the utility increment in a graphic way to show how the utility increases from underlying itemset to derivative itemset in Fig. 3.6. The utility increment is valued based on the same four datasets as the above section. The patterns are ordered by the utility of the DIs. The performance of our algorithm varies from one dataset to another. The performance in chainstore is much better than that in retail because the transaction time in chainstore (Fig. 3.6(b)) is 12 times more than that in retail (Fig. 3.6(a)) while the item types are only twice more. However, in the synthetic datasets in Fig. 3.6(d) and Fig. 3.6(c), the performance is much better. In conclusion, for each dataset, the performance is different but the utility actually increases from that of Underlying Itemset to Derivative Itemset.

3.4.4 Discussions

Based on the above datasets and experimental results, a table is used to demonstrate the conclusion that comes from our experiments and this is shown as Table 3.8. This table describes the number of itemsets whose utilities are increased or decreased within a given threshold. Also, two kinds of utility incremental forms are listed. One is the utility of derivative itemset which is higher than both the utilities of the underlying itemset and the additional itemset, which is denoted as FA; the other is that the utility of the derivative itemset is higher than the utility of the underlying itemset, which is denoted as FB. As for each underlying itemset, only one derivative itemset can be discovered, and some FA and FB might be ignored.

For the utility decrement itemsets whose utilities are only lower than the underlying itemsets, these are not considered in this table because these itemsets can also be regarded as FBs when the underlying itemsets and additional itemsets exchange.

In Table 3.8, Min_Sup in the minimum support for mining itemsets. $N.DI$ is the number of derivative itemsets discovered within the threshold



(a) retail

(b) chainstore

(c) t20i6d100k

(d) c20d10k

Figure 3.6: The experiment utility results of FUG comparing with F and U

Table 3.8: Utility Variation Conclusion

Dataset	Min_Sup	N.DI	R.U	N.FA	N.FB	N.DecI
Retail	0.01	89	20.3% - 50.4%	28	22	39
	0.008	135	18.6% - 50.4%	37	46	52
	0.002	1667	8.4% - 50.4%	473	769	425
Chainstore	0.002	79	4.6% - 207.2%	7	19	53
t20i6d100k	0.017	33	25.7% - 78.5%	8	11	14
	0.015	79	22.8% - 78.5%	24	19	36
	0.012	383	1.8% - 78.5%	112	137	134
c20d10k	0.05	120	19.7% - 150.9%	28	48	44

Min_Sup. $R.U$ in the utility incremental rate from the underlying itemset to the derivative itemset. $N.FA$ is the number of FA itemsets, $N.FB$ is the number of FB itemsets, and $N.DecI$ is the number of decremental itemsets.

3.5 Summary

Traditional high utility itemset mining methods possess the weakness that if the minimum utility threshold is set too high, the itemsets discovered may contain unrepresentative items; while if the threshold is set too low, too many redundant itemsets will be found. In addition, traditional association rule mining ignores the utility hidden among the items. To select both the high utility and high co-occurrence patterns, we propose an effective approach for identifying actionable combined utility itemsets. Furthermore, for different clusters of itemsets, only one itemset will be selected with the highest association-utility growth value, which caters for both high association and high utility. Thus, only the most effectively impacted itemsets will be presented. The results demonstrate that our method can discover patterns that are composed of different item combinations for both the utility increment and high representativeness. We will continue by working on mining novel combined utility itemsets and sequential patterns.

Chapter 4

Mining Strongly Associated and High Utility Incremental Patterns

4.1 Introduction

In this chapter, we present strongly associated and high utility incremental patterns.

As it has been proved that there exists a large gap between academic research topics and real world business and industrial challenges (Cao et al. 2010), the algorithms and approaches designed for academic purposes do not always satisfy real user needs in terms of solving business and industrial problems (Cao et al. 2007, Cao et al. 2008, Cao et al. 2010). Although algorithms such as UP-growth (Tseng et al. 2010), CHUD (Wu et al. 2011) and d^2HUI (Liu et al. 2012) can discover the complete set of high utility itemsets efficiently, they usually generate a large number of patterns, many of which are redundant and useless for making decision-oriented strategies. Real valuable patterns that managers or retailers might be interested in may be ignored or buried amongst thousands of similar patterns. In a word, they are not actionable.

Accordingly, the concept of actionable knowledge discovery is proposed by Cao (Cao et al. 2010, Cao et al. 2011) to select patterns that deliver decision-making informative massages to managers and retailers. The process of mining such patterns is called discovering actionable patterns. Instead of mining those patterns of high frequency, the actionable knowledge discovery approach mines patterns of high probability for decision-making purposes and impact-targeted activities are discovered. Such patterns may not be frequent but may lead to either impact-oriented exceptional behaviours or impact-contrasted exceptional behaviours.

For business purposes, retailers or supermarket managers seek products that are more profitable than others, thus they make combined product strategies. Such combined products might not be of high utility or frequency, but the additional products may hold a strong association with the underlying product which can increase profitability. Thus they cannot be selected via academic approaches, which focuses on discovering frequent or high utility patterns. This is also a challenge for designing an actionable combined pattern mining framework especially in this work.

Designing an actionable representation of patterns to satisfy both utility and frequency criteria is a challenging task and extracting such patterns is even harder. The contributions of our proposed method include:

- A combined framework for discovering actionable knowledge from a utility database. Based on a series of novel definitions such as *utility growth* which have never been used in previous work, we theoretically prove that the proposed representation is effective and the patterns discovered are really actionable and useful for decision-making.

- An efficient algorithm called MHUSAP(Mining High Utility and Strong Association Patterns) is proposed for mining actionable patterns of high utility increment and high dependence. We systematically analyse the relationship between underlying itemsets and derivative itemsets based on both utility and frequency criteria.

- Two effective strategies are applied to enhance the performance of MHUSAP. Based on the framework, we propose a global pruning strategy to generate utility increase patterns from underlying itemsets. An equation is also proposed as a local approach for selecting patterns with the highest weighted values, which is a hybrid measurement of both utility growth and dependence calculation.

The rest of this chapter is organised as follows. In section 2, we present two tables to describe some preliminaries and an extended comparison table is constructed based on an itemset's utility, support and confidence. In section 3, we demonstrate the problem in this chapter including a case study and an abstract combined model. In section 4, we mainly talk about our MHUSAP approach in terms of two strategies. A global utility growth tree structure is proposed to select all the utility incremental candidates, and a local *Coe* strategy is proposed to select one actionable pattern among a cluster of candidates composed of the same underlying itemsets. In section 5 we show some experimental results and we have a smart summary in section 6.

4.2 Preliminaries

Taking Table 4.1 and Table 4.2 as an example, let $D = \{T_1, T_2, ..., T_m\}$ be a transaction database, and $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ be a set of finite and discriminative items. Each transaction $T_c \in D$ $(1 < c < m)$ is a subset of $\mathcal{I}$ with a distinct identifier called *TID*. A transaction record can be regarded as a utility itemset. In a given transaction T_c, each item i_k appearing with a positive integer, $q(i_k, T_c)$ is called i_k's *quantity utility* in T_c. Also, each item in $\mathcal{I}$ is associated with a positive number $p(i_k, \mathcal{I})$, which is called i_k's *profit utility* in $\mathcal{I}$. Unlike sequential itemset or data stream, the items in an itemset are disordered, we assume that utility items are listed in alphabetical order without a loss of generality. An itemset contains l discriminative items called an *l-length* itemset, where $X \subseteq \mathcal{I}$. Also, an item is a 1-length itemset.

Table 4.1 shows a utility database containing five transactions with their

Table 4.1: Database Sample

TID	Transaction	TU
T_1	(A, 1) (C, 1) (D, 1)	8
T_2	(A, 1) (B, 6) (C, 2) (F, 5)	24
T_3	(A, 2) (B, 2) (C, 6) (D, 5) (E, 1)	60
T_4	(B, 4) (C, 3) (D, 2) (F, 3)	18
T_5	(B, 2) (C, 2) (F, 3)	9

Table 4.2: Profit Table of the Sample Database

Item	A	B	C	D	E	F
Profit	5	2	1	2	30	1

transaction utilities. Take the first transaction as an example, this utility itemset is a 3-length itemset because it contains 3 utility items {A}, {C} and {D}. The quantity utility of item {A} is 1 and presented as (A, 1). The Transaction ID (TID) is denoted as T_i in the ith transaction. Table 4.2 illustrates the unit utility (or called profit) of each item. One thing that should be noted is that the unit utility of item {E} is extremely high.

Definitions of *Frequency* and *Utility* are the same as demonstrated in Chapter 3 which are not listed here.

Given two itemsets, $X_o = \{i_{o_1}, i_{o_2}, ..., i_{o_m}\}$ and $X_a = \{i_{a_1}, i_{a_2}, ..., i_{a_n}\}$, X_a contains X_o if and only if there exists integers $1 \leq j_1 \leq j_2 \leq ... \leq j_m \leq n$ such that $i_{o_k} = i_{a_{j_k}}$ for $1 \leq k \leq m$, denoted as $X_o \subseteq X_a$. This containment concept is also suitable for utility itemsets with the same denotation. And also X_o is a sub-(utility-)itemset of X_a and X_a is the super-(utility-)itemset

of X_o.

For example, transaction T_4 contains 4 utility items {B, C, D and F} with the quantity utility of {4, 3, 2 and 3} separately. From table 4.2, the unit profit of each utility item is {B, 2}, {C, 1}, {D, 2} and {F, 1}, thus the final utility itemset composed of each utility item in transaction 4 can be presented as $T_4 = \{(B, 8) (C, 3) (D, 4) (F, 3)\}$ and the transaction utility of this itemset is 18. In addition, T_5 is a sub-itemset of both T_2 and T_4 and its transaction utility is 9. Furthermore, the transaction weighted utility of itemset {B, C, F} is 51 because it appears in three transactions T_2, T_4 and T_5.

Definition 4.1 *In equation 4.1, the itemset U is called an* ***Underlying Itemset****, which keeps invariant in all equations.* V_i *is called an* ***Additional Itemset*** *and* T_i *is called a* ***Derivative Itemset****.*

The roles that underlying itemsets and additional itemsets play can be exchanged in some cases, and the results may be different, which will be discussed in the next section.

$$C : \begin{cases} U + V_1 \rightarrow T_1 \\ U + V_2 \rightarrow T_2 \\ ... \\ U + V_n \rightarrow T_n \end{cases} \tag{4.1}$$

4.3 Problem Statement

4.3.1 Case Study

With the examples in the above tables, Table 4.3 lists the utility, support and confidence of itemsets (here each rule is also regarded as an itemset while rule $\{A \rightarrow E$ and rule $\{B \rightarrow A\}$ are one same itemset).

As is proved, the support of an itemset reduces when the itemset size (length) increases because of the *Downward Closure Property (DCP)*(Agrawal

Table 4.3: An Extended Comparison Table of Itemset Utility, Support and Confidence

Itemset(Pattern)	Utility	Support	Confidence
{A}	20	60%	Nil
{A → E}	40	20%	33%
{E}	30	20%	Nil
{E → A}	40	20%	100%
{C}	14	100%	Nil
{C → B}	41	80%	80%
{C → BE}	40	20%	20%

et al. 1994). However, some patterns still contain more information than others.

On one hand, business sale strategies can be made based on a given underlying itemset and several related additional itemsets, and a best strategy will be generated from those candidates. For example, $\{C\}$ is an itemset with a utility of 14 and support of 100%. Association rule $\{C \rightarrow B\}$ not only possesses a high confidence, say of 80%, but is also associated with a high utility increment from {C} (with utility 14) to {BC} (41). By contrast, itemset $\{B, C, E\}$ is associated with a high utility increment from 14 to 40, but the association rule $\{C \rightarrow BE\}$ is of a low confidence of 20%, which means customers might not to purchase $\{B, E\}$ after they purchase C.

On the other hand, variety underlying itemsets with the same additional itemset lead to different analysis results. For example, $\{A, E\}$ is an itemset with a utility of 40 and a support of 20% and it can generate two rules

$\{A \rightarrow E\}$ and $\{E \rightarrow A\}$ with different confidences: 33% and 100%. In addition, two component items $\{A\}$ and $\{E\}$ with their attributes are also listed in the table. To demonstrate this from a business perspective, when a retailer wants to make a promotion sale for product A, if he sells it with product E, the customer has a probability of 33% to purchase this product, but the profit of this product package will double from say 20 to 40. By contrast, when this retailer wants to make a promotion sale for product E with the same strategy, the customer might be highly likely to purchase A at the same time that E is purchased and the profit of this package thereby increases from 30 to 40.

4.3.2 An Abstract Model: A Representative Combined Pattern with Utility Increment and Strong Association

Here we illustrate the application of *actionable Combined Patterns* through identifying the *Representative Combined Pattern* from the Combined Representative Rule Cluster, which is shown in Equation 4.2.

$$C : \begin{cases} X_0 + X_1 \rightarrow X_a \\ X_0 + X_2 \rightarrow X_b \\ ... \end{cases} \tag{4.2}$$

Once a cluster of rules have been mined with combinations of the same underlying itemset and several additional itemsets, we begin to select the most interesting pattern among all the rules. This pattern has a balance of both the high utility increment from the underlying itemset to the derivative itemset and high association between the underlying itemset and the additional itemset. In the next section, we first propose a global strategy to discover all the utility increment patterns in the original database, then we demonstrate a local strategy of mining the most interesting combined pattern from a cluster of rules.

4.4 The MHUSAP Approach

In this section, we mainly focus on Mining High Utility Increment and Strong Association Patterns (MHUSAP). Our approach is based on two strategies: 1) One global strategy is proposed to discover all utility incremental patterns with clusters based on specific underlying itemsets; 2) One local strategy based on each cluster of patterns is proposed to measure the interestingness of each of the rules. This strategy is composed of a variety of additional itemsets and the same underlying itemset.

4.4.1 Mining Global Utility Incremental Itemsets Based on UG-Tree

UG-Tree

Our algorithm is based on utility growth and association rule mining, thus all branches with decreasing utilities should be pruned from the proposed UG-Tree. Since the utility growth of each node in a given branch could be either positive or negative, we prune branches from its external node until the first node whose utility growth becomes positive. This reorganised tree structure is called the *UG-Tree*, as shown in Fig. 4.1.

The composition of each node N is listed in Table 4.4. The *Rebuilt Table* is displayed to demonstrate the traversal of a UG-Tree. In the downward table, each row is composed of an *item name*, a *transaction-weighted utilisation value* and a *link*. Each link points to the node having the same item name as shown in the UG-Tree. The nodes with the same item names can be traversed efficiently by following the links between the downward table and the nodes in the UG-Tree.

UG-Tree Construction

A UG-Tree can be constructed by scanning the original database only twice. During the first database scan, items and their TWUs are captured via the

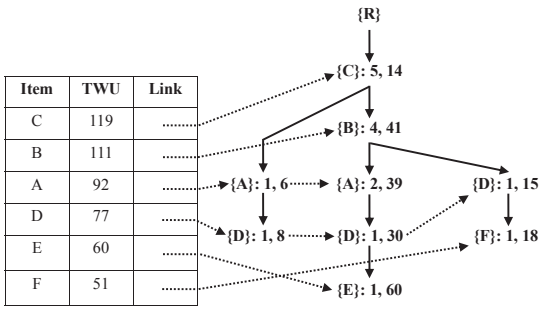


Figure 4.1: Downward table and a UG-Tree when *min_util = 0*

Table 4.4: Composition in Each Node

Formation	Explanation
N.na	the item name of this node
N.sc	the support count of this node
N.u	the node utility of this node
N.p	the parent node of this node
N.nl	the node link which points to a node whose name is same as N.na

calculation of the transaction utility of each transaction and the transaction weighted utilisation of each item. The downward table is then formed by items inserted in the TWU-decreasing order. In the second scan, the reorganised transactions as well as their reorganised transaction utilities are inserted into the UG-Tree, as shown in Table 3.5. In addition, all utility-decrement branches are pruned and shown in Fig. 4.1. Finally, an UG-Tree

is created with root R.

Definition 4.2 *A transaction after the process of the above reorganisation is called a* Rebuilt Transaction *and its TU is called a* rebuilt transaction utility (RTU*), denoted as* $RTU(T_c)$.

The construction of the UG-Tree will be completed after all RTs are inserted with their RTUs. Fig. 4.1 presents a UG-Tree when the minimum utility threshold is set at zero.

By using the UG-Tree, which contains enough information, we can generate the utility-association rules. While the UG-Tree is retrieved, the utility of each itemset can be discovered and prepared for the calculation of *Factor C*. At the same time, the support count of each item can also be found through *item.nl* and used for *Factor W*. These two factors will be discussed in the proceeding subsections.

4.4.2 Mining Locally Interesting Patterns from Clusters of Patterns

Based on all the global utility incremental itemsets discovered by the UG-Tree, we reorganise the itemsets and form clusters of actionable combined patterns. Each cluster is composed of several derivative itemsets which are also combined patterns including one underlying itemset as well as several additional itemsets and these are described as Equation 4.2.

Since there exists weakness in any single-measurement-based framework, we propose a combined mining approach to discover those really interesting patterns in business and industrial areas. To measure the interestingness of each combined pattern in order to select the most actionable one, we propose two parameters to value both the utility increment and the association relationship. Furthermore, we propose a coefficient to balance the importance of two values, Thus, in one cluster of derivative itemsets, the actionable pattern is a combined pattern with the highest coefficient value.

The Impact of Additional Itemset to the Underlying Itemset

Definition 4.3 *The* impact *of the Additional Itemset (ΔX) is proposed as an impact factor to value the utility growth within each actionable combined pattern, specifically, from the utility of the underlying itemset to the derivative itemset. It is denoted as $I(\Delta X \rightarrow X_0)$, and defined as:*

$$I(\Delta X \rightarrow X_0) = \frac{U(X)}{U(X_0)} \tag{4.3}$$

This equation is proposed to measure the incremental degree of the derivative itemset from the underlying itemset and from the utility perspective. Here, $U(X)$ is the utility of derivative itemset X in the whole database.

the Confidence of the Additional Itemset to the Underlying Itemset

Definition 4.4 *The* confidence *of the additional itemset to measure the association of the additional itemset to that of the underlying itemset, denoted as $C(\Delta X \rightarrow X_0)$, is defined as:*

$$C(\Delta X \rightarrow X_0) = \frac{Supp(X)}{Supp(X_0} \tag{4.4}$$

This equation is proposed to examine whether the additional itemset ΔX has a strong association with the underlying itemset X_0. Here, $Supp(X)$ is the support of derivative itemset X, and $Supp(X_0)$ is the support of underlying itemset X_0. In addition,

$$Supp(X) = Supp(\Delta X \rightarrow X_0) \tag{4.5}$$

Combined Coefficient of Additional Itemset to Underlying Itemset

Definition 4.5 *The* combined coefficient *of the additional itemset is proposed to measure the interestingness that is the degree of the effectiveness of manufacturing the derivative itemset from the underlying itemset, denoted as*

$CoE(\Delta X \rightarrow X_0)$, *defined as:*

$$CoE(\Delta X \rightarrow X_0) = \sqrt{I(\Delta X \rightarrow X_0) * C(\Delta X \rightarrow X_0)} \tag{4.6}$$

This equation averages the value of $I(\Delta X \rightarrow X_0)$ in Equation 4.3 and $C(\Delta X \rightarrow X_0)$ in Equation 4.4. It is easy to see that even though the utility from the underlying itemset to the derivative itemset increases a lot, it makes no sense if the association between the underlying itemset and the additional itemset is weak, which means customers might not purchase such products together in most circumstances.

In addition, for a specific underlying itemset X_0, the higher CoE means that this itemset or product package is likely to attract customers and thus will be purchased together. Among all additional itemsets, only the most interesting will be selected due to our approach.

4.5 Experimental Results

In this section, we conduct intensive experiments to evaluate the proposed methods. Our experiments are run on a PC with a 2.30 GHz Intel Core and a 16 gigabyte memory. MHUSAP is implemented in Java. Two real datasets and two synthetic datasets are used for the experiments. The real datasets are *Retail*[1] and *Chainstore*[2], and the synthetic datasets are *t20i6d100k* and *c20d10k*. The parameters of the datasets are listed in Table 4.5.

4.5.1 Candidates Generated by Different Thresholds

In our experiment, the candidates are first generated by a global pruning strategy via the UG-Tree. Then, based on the different thresholds, various patterns are discovered. We do not set the specific confidence threshold or the impact threshold. Only one parameter threshold is to be considered

[1] `http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php`
[2] `http://cucis.ece.northwestern.edu/projects/DMS/MineBench.html`

Table 4.5: Features of the Datasets

Dataset	Number of Transactions	Number of Items	Average Length
Retail[3]1	88162	16470	10.3
Chainstore[4]2	1112949	46086	7.3
t20i6d100k	100000	658	13.7
c20d10k	10000	187	13

which is the CoE. The numbers of candidates and patterns discovered by each process are listed in table 4.6.

4.5.2 Utility Incremental Figures

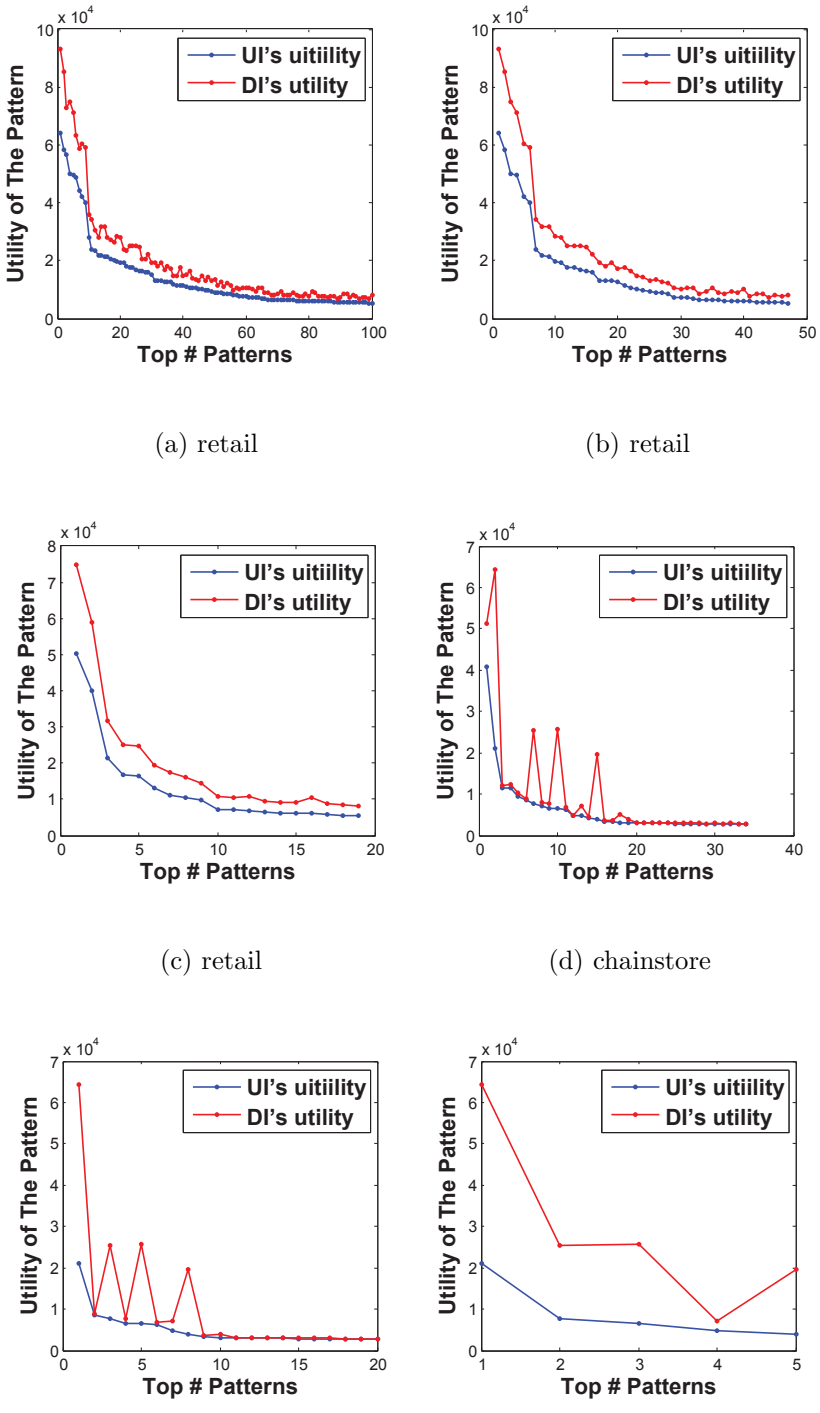
We also plot some figures to show the utility increment of the patterns, specifically, from the underlying itemset to the derivative itemset. Two real datasets are applied with different CoE thresholds. Firstly three figures are extracted from retail with the threshold 1.2, 1.15 and 1.1. Secondly, three figures are selected from chainstore, with the threshold 1.2, 0.9 and 0.5. The patterns are ordered by the utility decrement of the underlying itemsets.

4.6 Summary

Since there exists a big gap between the academic outcomes and business or industrial purposes in the data mining domain, common high utility or frequent patterns might be of no interest to the business world. In addition, those rare or low utility items might attract customers because of their representativeness. This chapter proposes an alternative approach to discovering patterns with strong association and high utility increment, which have proved to be useful. One efficient tree structure is proposed as a global

Table 4.6: Utility Variation Summarization

Dataset: Retail	UG-Tree	TH0.6	TH1.1	TH1.15	TH1.2
Number of Candidates	2408	937	100	47	19
Dataset: ChainStore	UG-Tree	TH0.5	TH0.7	TH0.9	TH1.2
Number of Candidates	1150	34	29	22	5
Dataset: t20i6d100k	UG-Tree	TH1.0	TH1.1	TH1.2	TH1.5
Number of Candidates	3432	1484	410	67	10
Dataset: c20d10k	UG-Tree	TH1.4	TH1.5	TH0.9	TH2.0
Number of Candidates	2251	435	126	32	7



(a) retail (b) retail

(c) retail (d) chainstore

(e) chainstore (f) chainstore

Figure 4.2: Experiments for utility incremental on real datasets

strategy to select these utility increment itemsets. In addition, two factors used to measure both utility and frequency are proposed. Finally, the combined coefficient is discussed to measure the interestingness of each cluster of derivative patterns.

During the experiment process, an interesting pattern was discovered in that the derivative itemset can also be regarded as a new underlying itemset in the chainstore dataset. This might be an interesting area to explore in the future.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In conclusion, this thesis presents several techniques to address Actionable Knowledge Discovery based on combined mining approaches with both frequency and utility. Actionable knowledge discovery has become an increasingly important issue in data mining and machine learning domain in recent years. Traditional pattern mining focuses on extracting patterns in the frequency/support framework or utility framework which does not have business or industrial value and impact, thus it is not actionable or useful for business decision-making. In addition, the introduction of "actionable" has not only brought decision-making oriented knowledge to the pattern mining domain, but it has also brought new problems. First, the absence of the quality (profit) and quantity of the item as applied in the utility mining area makes the mining results much less convincing. Patterns abstracted from the frequency/support framework are not suitable for every business model. Novel structures and algorithms need to be designed to improve the modeling process and performance. Second, according to our experience in the mining utility database, the extracted patterns as well as the candidates suffer from a large amount of redundancy which is similar to frequent pattern mining. Many patterns look quite similar to one another because common items exist

in several of the patterns. The challenge here is to explore approaches and algorithms which can efficiently and effectively summarise the patterns and select the most interesting ones among a cluster of similar patterns.

In chapter 3, we present a novel combined structure for selecting those patterns with both high utility increment and high co-occurrence. This is because traditional high utility itemset mining methods often ignore the frequency of each itemset, and frequent patterns are also not actionable to retailers and supermarket managers. In addition, in order to overcome the weak points in the utility mining process (i.e. if the minimum utility threshold is set too high, the itemsets discovered might contain unrepresentative items; while if the threshold is set too low, too many redundant itemsets will be found), we propose an effective approach for identifying actionable combined utility itemsets. For different clusters of itemsets, only one itemset will be selected with the highest association-utility growth value, which caters for both high association and high utility growth. Thus, only the most effectively impacted itemsets will be presented. To tackle this issue, we provide a systematic statement of a generic framework which defines calculations of the utility growth from the underlying itemset to the derivative itemset and the co-occurrence frequency of each underlying itemset and additional itemset. We specify a naive combination of these two factors with a quadratic mean to select both high utility and high frequency patterns. The effective algorithm CUARM is proposed and the results based on both the real datasets and synthetic datasets demonstrate that our method can discover patterns that are composed of different item combinations of both the utility increment and high representativeness.

In chapter 4, we propose an efficient algorithm named MHUSAP, which is short for Mining High Utility and Strong Association Patterns. Since there exists a big gap between academic outcomes and business or industrial objectives in the data mining domain, high utility or frequent patterns might be very common and therefore of little interest to businessman or woman. In addition, those rare or low utility items might attract too much interest due

to their representativeness. This chapter propose an alternative approach to discover those patterns with strong association and high utility increment, which have proved to be useful. One efficient tree structure is proposed as a global strategy to select utility increment itemsets. In addition, two factors used to measure both utility and frequency are proposed. Finally, the combined coefficient is discussed to measure the interestingness of each cluster of derivative patterns. The experimental results show that this algorithm works as we expect.

Each chapter (i.e. Chapter 3 and Chapter 4) of this thesis is supported by one published conference papers[1] listed in the **List of Publications**. Therefore, what we have done and proposed in this thesis is of great significance to the Actionable Combined Knowledge Discovery for business and industrial areas.

5.2 Future Work

Actionable combined knowledge discovery is very promising but it is still in the initial stages. Extensive work needs to be undertaken to finalise the grand framework and to extend to novel combined domains. This work includes:

(i). **Actionable Combined High Utility Sequential Patterns**: We have implemented the combination of frequent pattern and high utility patterns. Furthermore, high sequential utility pattern mining can be regarded as an extension of high utility itemset mining, thus it can be explored more in the future.

(ii). **Actionable High Utility Pattern Chains**: during the MHUSAP experimental process, we noticed some interesting patterns i.e. the derivative itemset can also be regarded as a new underlying itemset in the chainstore dataset. This phenomenon might become an interesting new area and could be explored so that a chain of patterns can be

[1]The papers of chapter 3 is published, the paper of chapter 4 is still under review

selected in the future. In addition, a l-length pattern is composed of a number of "l-1" underlying itemsets and "l-1" derivative itemsets. We can also select the best utility curve from the increase in pattern length.

Bibliography

Agrawal, R. & Shafer, J. C. (1996), 'Parallel mining of association rules', *IEEE Transactions on Knowledge & Data Engineering* (6), 962–969.

Agrawal, R. & Srikant, R. (1995), Mining sequential patterns, *in* 'Data Engineering, 1995. Proceedings of the Eleventh International Conference on', IEEE, pp. 3–14.

Agrawal, R., Srikant, R. et al. (1994), Fast algorithms for mining association rules, *in* 'Proc. 20th int. conf. very large data bases, VLDB', Vol. 1215, pp. 487–499.

Ahmed, C. F., Tanbeer, S. K. & Jeong, B.-S. (2010*a*), Efficient mining of high utility patterns over data streams with a sliding window method, *in* 'Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2010', Springer, pp. 99–113.

Ahmed, C. F., Tanbeer, S. K. & Jeong, B.-S. (2010*b*), Mining high utility web access sequences in dynamic web log data, *in* 'Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD), 2010 11th ACIS International Conference on', IEEE, pp. 76–81.

Ahmed, C. F., Tanbeer, S. K. & Jeong, B.-S. (2010*c*), 'A novel approach for mining high-utility sequential patterns in sequence databases', *ETRI journal* **32**(5), 676–686.

Ahmed, C. F., Tanbeer, S. K., Jeong, B.-S. & Choi, H.-J. (2012), 'Interactive mining of high utility patterns over data streams', *Expert Systems with Applications* **39**(15), 11979–11991.

Ahmed, C. F., Tanbeer, S. K., Jeong, B.-S. & Lee, Y.-K. (2009), 'Efficient tree structures for high utility pattern mining in incremental databases', *Knowledge and Data Engineering, IEEE Transactions on* **21**(12), 1708–1721.

Ahmed, C. F., Tanbeer, S. K., Jeong, B.-S., Lee, Y.-K. & Choi, H.-J. (2012), 'Single-pass incremental and interactive mining for weighted frequent patterns', *Expert Systems with Applications* **39**(9), 7976–7994.

Bayardo Jr, R. J. (1998), Efficiently mining long patterns from databases, *in* 'ACM Sigmod Record', Vol. 27, ACM, pp. 85–93.

Birkhoff, G. (1967), 'Lattice theory amer', *Math. Soc, Providence, RI* .

Burdick, D., Calimlim, M. & Gehrke, J. (2001), Mafia: A maximal frequent itemset algorithm for transactional databases, *in* 'Data Engineering, 2001. Proceedings. 17th International Conference on', IEEE, pp. 443–452.

Cai, C. H., Fu, A. W., Cheng, C. & Kwong, W. (1998), Mining association rules with weighted items, *in* 'Database Engineering and Applications Symposium, 1998. Proceedings. IDEAS'98. International', IEEE, pp. 68–77.

Cao, L. (2012), 'Combined mining: Analyzing object and pattern relations for discovering actionable complex patterns', *sponsored by Australian Research Council Discovery Grants (DP1096218 and DP130102691) and an ARC Linkage Grant (LP100200774)* .

Cao, L., Philip, S. Y., Zhang, C. & Zhao, Y. (2010), *Domain driven data mining*, Springer.

Cao, L., Zhang, H., Zhao, Y., Luo, D. & Zhang, C. (2011), 'Combined mining: discovering informative knowledge in complex data', *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **41**(3), 699–712.

Cao, L., Zhao, Y., Figueiredo, F., Ou, Y. & Luo, D. (2007), Mining high impact exceptional behavior patterns, *in* 'Emerging Technologies in Knowledge Discovery and Data Mining', Springer, pp. 56–63.

Cao, L., Zhao, Y. & Zhang, C. (2008), 'Mining impact-targeted activity patterns in imbalanced data', *Knowledge and Data Engineering, IEEE Transactions on* **20**(8), 1053–1066.

Chan, R. C., Yang, Q. & Shen, Y.-D. (2003), Mining high utility itemsets, *in* 'Data Mining, 2003. ICDM 2003. Third IEEE International Conference on', IEEE, pp. 19–26.

Chang, J. H. (2011), 'Mining weighted sequential patterns in a sequence database with a time-interval weight', *Knowledge-Based Systems* **24**(1), 1–9.

Cheung, D. W., Han, J., Ng, V. T., Fu, A. W. & Fu, Y. (1996), A fast distributed algorithm for mining association rules, *in* 'Parallel and Distributed Information Systems, 1996., Fourth International Conference on', IEEE, pp. 31–42.

Cheung, D. W., Han, J., Ng, V. T. & Wong, C. (1996), Maintenance of discovered association rules in large databases: An incremental updating technique, *in* 'Data Engineering, 1996. Proceedings of the Twelfth International Conference on', IEEE, pp. 106–114.

Cheung, W. & Zaiane, O. R. (2003), Incremental mining of frequent patterns without candidate generation or support constraint, *in* 'Database Engineering and Applications Symposium, 2003. Proceedings. Seventh International', IEEE, pp. 111–116.

Cheung, Y.-L. & Fu, A. W.-C. (2004), 'Mining frequent itemsets without support threshold: with and without item constraints', *Knowledge and Data Engineering, IEEE Transactions on* **16**(9), 1052–1069.

Erwin, A., Gopalan, R. P. & Achuthan, N. (2007*a*), A bottom-up projection based algorithm for mining high utility itemsets, *in* 'Proceedings of the 2nd international workshop on Integrating artificial intelligence and data mining-Volume 84', Australian Computer Society, Inc., pp. 3–11.

Erwin, A., Gopalan, R. P. & Achuthan, N. (2007*b*), Ctu-mine: an efficient high utility itemset mining algorithm using the pattern growth approach, *in* 'Computer and Information Technology, 2007. CIT 2007. 7th IEEE International Conference on', IEEE, pp. 71–76.

Erwin, A., Gopalan, R. P. & Achuthan, N. (2008), Efficient mining of high utility itemsets from large datasets, *in* 'Advances in Knowledge Discovery and Data Mining', Springer, pp. 554–561.

Fournier-Viger, P., Wu, C.-W., Zida, S. & Tseng, V. S. (2014), Fhm: Faster high-utility itemset mining using estimated utility co-occurrence pruning, *in* 'Foundations of intelligent systems', Springer, pp. 83–92.

Grahne, G. & Zhu, J. (2003), Efficiently using prefix-trees in mining frequent itemsets., *in* 'FIMI', Vol. 90.

Han, J., Cheng, H., Xin, D. & Yan, X. (2007), 'Frequent pattern mining: current status and future directions', *Data Mining and Knowledge Discovery* **15**(1), 55–86.

Han, J., Pei, J. & Yin, Y. (2000), Mining frequent patterns without candidate generation, *in* 'ACM SIGMOD Record', Vol. 29, ACM, pp. 1–12.

Han, J., Wang, J., Lu, Y. & Tzvetkov, P. (2002), Mining top-k frequent closed patterns without minimum support, *in* 'Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on', IEEE, pp. 211–218.

Hong, T.-P., Wang, C.-Y. & Tao, Y.-H. (2001), 'A new incremental data mining algorithm using pre-large itemsets', *Intelligent Data Analysis* **5**(2), 109–128.

Khan, M. S., Muyeba, M. & Coenen, F. (2008), A weighted utility framework for mining association rules, *in* 'Computer Modeling and Simulation, 2008. EMS'08. Second UKSIM European Symposium on', IEEE, pp. 87–92.

Koh, J.-L. & Shieh, S.-F. (2004), An efficient approach for maintaining association rules based on adjusting fp-tree structures, *in* 'Database Systems for Advanced Applications', Springer, pp. 417–424.

Lan, G.-C., Hong, T.-P. & Chao, Y.-T. (2014*a*), Multi-criteria utility mining using maximum constraints, *in* 'Computational Collective Intelligence. Technologies and Applications', Springer, pp. 466–471.

Lan, G.-C., Hong, T.-P. & Chao, Y.-T. (2014*b*), Multi-criteria utility mining using minimum constraints, *in* 'Modern Advances in Applied Intelligence: 27th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2014, Kaohsiung, Taiwan, June 3-6, 2014, Proceedings', Vol. 8482, Springer, p. 42.

Lan, G.-C., Hong, T.-P., Huang, H.-C. & Pan, S.-T. (2013), Mining high fuzzy utility sequential patterns, *in* 'Fuzzy Theory and Its Applications (iFUZZY), 2013 International Conference on', IEEE, pp. 420–424.

Lee, G. & Yun, U. (2012), Mining weighted frequent sub-graphs with weight and support affinities, *in* 'Multi-disciplinary Trends in Artificial Intelligence', Springer, pp. 224–235.

Li, H.-F., Huang, H.-Y., Chen, Y.-C., Liu, Y.-J. & Lee, S.-Y. (2008), Fast and memory efficient mining of high utility itemsets in data streams, *in*

'Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on', IEEE, pp. 881–886.

Li, H.-F., Huang, H.-Y. & Lee, S.-Y. (2011), 'Fast and memory efficient mining of high-utility itemsets from data streams: with and without negative item profits', *Knowledge and information systems* **28**(3), 495–522.

Li, Y.-C. & Yeh, J.-S. (2005), C.: Efficient algorithms for mining share-frequent itemsets, *in* 'In Proceedings of the 11th World Congress of Intl. Fuzzy Systems Association', Citeseer.

Li, Y.-C., Yeh, J.-S. & Chang, C.-C. (2005), Direct candidates generation: a novel algorithm for discovering complete share-frequent itemsets, *in* 'Fuzzy Systems and Knowledge Discovery', Springer, pp. 551–560.

Li, Y.-C., Yeh, J.-S. & Chang, C.-C. (2008), 'Isolated items discarding strategy for discovering high utility itemsets', *Data & Knowledge Engineering* **64**(1), 198–217.

Lin, C.-W., Hong, T.-P., Lan, G.-C., Chen, H.-Y. & Kao, H.-Y. (2010), Incrementally mining high utility itemsets in dynamic databases, *in* 'Granular Computing (GrC), 2010 IEEE International Conference on', IEEE, pp. 303–307.

Lin, C.-W., Hong, T.-P., Lan, G.-C., Wong, J.-W. & Lin, W.-Y. (2013), Mining high utility itemsets based on the pre-large concept, *in* 'Advances in Intelligent Systems and Applications-Volume 1', Springer, pp. 243–250.

Lin, C.-W., Hong, T.-P., Lan, G.-C., Wong, J.-W. & Lin, W.-Y. (2014), 'Incrementally mining high utility patterns based on pre-large concept', *Applied intelligence* **40**(2), 343–357.

Lin, C.-W., Lan, G.-C. & Hong, T.-P. (2012), 'An incremental mining algorithm for high utility itemsets', *Expert Systems with Applications* **39**(8), 7173–7180.

Lin, X., Zhu, Q., Li, F., Geng, Z. & Shi, S. (2010), A share strategy for utility frequent patterns mining, *in* 'Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on', Vol. 3, IEEE, pp. 1428–1432.

Liu, G., Lu, H., Yu, J. X., Wang, W. & Xiao, X. (2003), Afopt: An efficient implementation of pattern growth approach., *in* 'FIMI'.

Liu, J., Wang, K. & Fung, B. (2012), Direct discovery of high utility itemsets without candidate generation, *in* 'Data Mining (ICDM), 2012 IEEE 12th International Conference on', IEEE, pp. 984–989.

Liu, M. & Qu, J. (2012), Mining high utility itemsets without candidate generation, *in* 'Proceedings of the 21st ACM international conference on Information and knowledge management', ACM, pp. 55–64.

Liu, Y., Liao, W.-k. & Choudhary, A. (2005), A fast high utility itemsets mining algorithm, *in* 'Proceedings of the 1st international workshop on Utility-based data mining', ACM, pp. 90–99.

Mannila, H., Toivonen, H. & Verkamo, A. I. (1995), Discovering frequent episodes in sequences extended abstract, *in* 'Proceedings the first Conference on Knowledge Discovery and Data Mining', pp. 210–215.

Park, J. S., Chen, M.-S. & Yu, P. S. (1995*a*), *An effective hash-based algorithm for mining association rules*, Vol. 24, ACM.

Park, J. S., Chen, M.-S. & Yu, P. S. (1995*b*), Efficient parallel data mining for association rules, *in* 'Proceedings of the fourth international conference on Information and knowledge management', ACM, pp. 31–36.

Pasquier, N., Bastide, Y., Taouil, R. & Lakhal, L. (1999), Discovering frequent closed itemsets for association rules, *in* 'Database Theory?ICDT?99', Springer, pp. 398–416.

Pei, J., Han, J., Mao, R. et al. (2000), Closet: An efficient algorithm for mining frequent closed itemsets., *in* 'ACM SIGMOD workshop on research issues in data mining and knowledge discovery', Vol. 4, pp. 21–30.

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U. & Hsu, M.-C. (2001), Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth, *in* 'icccn', IEEE, p. 0215.

Quang, T. M., Oyanagi, S. & Yamazaki, K. (2006), Exminer: an efficient algorithm for mining top-k frequent patterns, *in* 'Advanced Data Mining and Applications', Springer, pp. 436–447.

Savasere, A., Omiecinski, E. R. & Navathe, S. B. (1995), 'An efficient algorithm for mining association rules in large databases'.

Shie, B.-E., Philip, S. Y. & Tseng, V. S. (2012), 'Efficient algorithms for mining maximal high utility itemsets from data streams with different models', *Expert Systems with Applications* **39**(17), 12947–12960.

Shie, B.-E., Tseng, V. S. & Yu, P. S. (2010), Online mining of temporal maximal utility itemsets from data streams, *in* 'Proceedings of the 2010 ACM Symposium on Applied Computing', ACM, pp. 1622–1626.

Song, W., Liu, Y. & Li, J. (2012), Vertical mining for high utility itemsets, *in* 'Granular Computing (GrC), 2012 IEEE International Conference on', IEEE, pp. 429–434.

Song, W., Liu, Y. & Li, J. (2014), 'Mining high utility itemsets by dynamically pruning the tree structure', *Applied intelligence* **40**(1), 29–43.

Srikant, R. & Agrawal, R. (1996), *Mining sequential patterns: Generalizations and performance improvements*, Springer.

Sucahyo, Y. G. & Gopalan, R. P. (2003), Ct-itl: Efficient frequent item set mining using a compressed prefix tree with pattern growth, *in* 'Proceedings of the 14th Australasian database conference-Volume 17', Australian Computer Society, Inc., pp. 95–104.

Tanbeer, S. K., Ahmed, C. F., Jeong, B.-S. & Lee, Y.-K. (2008), Cp-tree: a tree structure for single-pass frequent pattern mining, *in* 'Advances in Knowledge Discovery and Data Mining', Springer, pp. 1022–1027.

Tao, F., Murtagh, F. & Farid, M. (2003), Weighted association rule mining using weighted support and significance framework, *in* 'Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 661–666.

Tseng, V. S., Chu, C.-J. & Liang, T. (2006), Efficient mining of temporal high utility itemsets from data streams, *in* 'Second International Workshop on Utility-Based Data Mining', Citeseer, p. 18.

Tseng, V. S., Shie, B.-E., Wu, C.-W. & Yu, P. S. (2013), 'Efficient algorithms for mining high utility itemsets from transactional databases', *Knowledge and Data Engineering, IEEE Transactions on* **25**(8), 1772–1786.

Tseng, V. S., Wu, C.-W., Shie, B.-E. & Yu, P. S. (2010), Up-growth: an efficient algorithm for high utility itemset mining, *in* 'Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 253–262.

Wang, J., Han, J. & Pei, J. (2003), Closet+: Searching for the best strategies for mining frequent closed itemsets, *in* 'Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 236–245.

Wang, W., Yang, J. & Yu, P. S. (2000), Efficient mining of weighted association rules (war), *in* 'Proceedings of the sixth ACM SIGKDD inter-

national conference on Knowledge discovery and data mining', ACM, pp. 270–274.

Wu, C. W., Fournier-Viger, P., Yu, P. S. & Tseng, V. S. (2011), Efficient mining of a concise and lossless representation of high utility itemsets, *in* 'Data Mining (ICDM), 2011 IEEE 11th International Conference on', IEEE, pp. 824–833.

Wu, C.-W., Lin, Y.-F., Yu, P. S. & Tseng, V. S. (2013), Mining high utility episodes in complex event sequences, *in* 'Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 536–544.

Wu, C. W., Shie, B.-E., Tseng, V. S. & Yu, P. S. (2012), Mining top-k high utility itemsets, *in* 'Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 78–86.

Yao, H. & Hamilton, H. J. (2006), 'Mining itemset utilities from transaction databases', *Data & Knowledge Engineering* **59**(3), 603–626.

Yao, H., Hamilton, H. J. & Butz, C. J. (2004), A foundational approach to mining itemset utilities from databases., *in* 'SDM', Vol. 4, SIAM, pp. 215–221.

Yeh, J.-S., Li, Y.-C. & Chang, C.-C. (2007), Two-phase algorithms for a novel utility-frequent mining model, *in* 'Emerging Technologies in Knowledge Discovery and Data Mining', Springer, pp. 433–444.

Yen, S.-J. & Lee, Y.-S. (2007), Mining high utility quantitative association rules, *in* 'Data warehousing and knowledge discovery', Springer, pp. 283–292.

Yin, J., Zheng, Z. & Cao, L. (2012), Uspan: an efficient algorithm for mining high utility sequential patterns, *in* 'Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 660–668.

Yin, J., Zheng, Z., Cao, L., Song, Y. & Wei, W. (2013), Efficiently mining top-k high utility sequential patterns, *in* 'Data Mining (ICDM), 2013 IEEE 13th International Conference on', IEEE, pp. 1259–1264.

Yun, U. (2007*a*), 'Efficient mining of weighted interesting patterns with a strong weight and/or support affinity', *Information Sciences* **177**(17), 3477–3499.

Yun, U. (2007*b*), 'Mining lossless closed frequent patterns with weight constraints', *Knowledge-Based Systems* **20**(1), 86–97.

Yun, U. (2008*a*), 'An efficient mining of weighted frequent patterns with length decreasing support constraints', *Knowledge-Based Systems* **21**(8), 741–752.

Yun, U. (2008*b*), 'A new framework for detecting weighted sequential patterns in large sequence databases', *Knowledge-Based Systems* **21**(2), 110–122.

Yun, U. & Leggett, J. J. (2005*a*), 'Wfim: Weighted frequent itemset mining'.

Yun, U. & Leggett, J. J. (2005*b*), Wlpminer: weighted frequent pattern mining with length-decreasing support constraints, *in* 'Advances in Knowledge Discovery and Data Mining', Springer, pp. 555–567.

Yun, U. & Leggett, J. J. (2006*a*), Wip: mining weighted interesting patterns with a strong weight and/or support affinity., *in* 'SDM', SIAM, pp. 624–628.

Yun, U. & Leggett, J. J. (2006*b*), Wspan: Weighted sequential pattern mining in large sequence databases, *in* 'Intelligent Systems, 2006 3rd International IEEE Conference on', IEEE, pp. 512–517.

Yun, U., Shin, H., Ryu, K. H. & Yoon, E. (2012), 'An efficient mining algorithm for maximal weighted frequent patterns in transactional databases', *Knowledge-Based Systems* **33**, 53–64.

Zaki, M. J. (2000), 'Scalable algorithms for association mining', *Knowledge and Data Engineering, IEEE Transactions on* **12**(3), 372–390.

Zaki, M. J. (2001), 'Spade: An efficient algorithm for mining frequent sequences', *Machine learning* **42**(1-2), 31–60.

Zaki, M. J. & Hsiao, C.-J. (2002), Charm: An efficient algorithm for closed itemset mining., *in* 'SDM', Vol. 2, SIAM, pp. 457–473.

Zaki, M. J., Parthasarathy, S., Ogihara, M. & Li, W. (1997), 'Parallel algorithms for discovery of association rules', *Data Mining and Knowledge Discovery* **1**(4), 343–373.

Zhang, H., Zhao, Y., Cao, L. & Zhang, C. (2008), Combined association rule mining, *in* 'Advances in Knowledge Discovery and Data Mining', Springer, pp. 1069–1074.

Zhao, Y., Zhang, H., Cao, L., Zhang, C. & Bohlscheid, H. (2008), Combined pattern mining: From learned rules to actionable knowledge, *in* 'AI 2008: Advances in Artificial Intelligence', Springer, pp. 393–403.

Zhao, Y., Zhang, H., Figueiredo, F., Cao, L. & Zhang, C. (2007), Mining for combined association rules on multiple datasets, *in* 'Proceedings of the 2007 international workshop on Domain driven data mining', ACM, pp. 18–23.

Zhou, L., Liu, Y., Wang, J. & Shi, Y. (2007), Utility-based web path traversal pattern mining, *in* 'Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on', IEEE, pp. 373–380.