Chapter 1

Introduction

1.1 Background

Frequent itemset mining (FIM) has been a major task in data mining. The aim of FIM is to discover the itemsets that frequently occur together. FIM is used to mine the most frequent items from the transactional dataset which fulfill the minimum support *min_sup* threshold, where *min_sup* is a parameter set by the user. The concept of FIM was introduced by Agrawal and Srikant [1]. Later, many algorithms [2, 3, 4, 5, 6] have been developed for mining frequent itemset. The main challenge in FIM is to develop a fast and efficient algorithm that can handle large volume of data, minimum time scans dataset and find rule very quickly. Most of the proposed Apriori-like algorithms for mining frequent itemset waste lots of time to generate candidate itemsets. FP-Growth algorithm [7] is also very useful for finding frequent itemset. It is faster than Apriori algorithm and consumes lesser memory. FP-Growth algorithm does not generate candidate itemsets so take less time to find frequent itemsets. However, it has limitation in respect of space and time. FIM algorithms assume that item cannot appear more than once in each transaction and each item has same importance like weight, unit profit, etc. Hiding importance and quantity of an item may also hide some important or relevant information. Hence, FIM not only loses valuable and important information of the itemsets but also generates many irrelevant and unimportant frequent itemsets. However, in real-life, retailers are interested to find the important itemsets rather than frequent itemsets. To address the issue of quantity,

multi-frequency based frequent itemset mining algorithm has been proposed [4], but only quantity based mining loose importance of the items.

In order to overcome these problems, high utility itemsets (HUIs) mining algorithms have been proposed [8, 9, 10]. In HUIs mining, items can have the quantity and relative importance. HUIs have numerous applications such as market basket analysis [9, 11, 12], website click stream [13, 14], cross-marketing in retail stores [15, 16, 17, 11], business intelligence [18, 19], biomedical applications [8] and mobile commerce applications [20, 21], etc.

1.2 Issues and Challenges

Chan et al. introduced the concept of utility based itemset mining in 2003 [8]. For the first time utility based mining method consider *utility* is a measure of how useful an itemset is. Later on, Yao et al. defined two types of utilities for items, transaction utility (internal utility) and external utility where transaction utility denotes the sold quantity of an item [9]. Both above-discussed methods are not efficient and mine all HUIs. There no any strategy is exist to mine rules efficiently. HUIs mining does not support the *downward closure property*[1] because the utility of an itemset may be smaller, equal or greater to the utility of its supersets (or subsets). Hence, HUIs mining is more difficult than FIM. Therefore, prune the search space in HUIs mining is the difficult task. To tackle this problem, Liu et al. [22] proposed the concept of transaction weighted utility (TWU) to facilitate the performance of the mining task. TWU is used for overestimation of true or actual utility of an itemset and hence, used as *downward closure property*. However, two-phase based model leads to waste lots of time to compute the utility for itemsets because their level-wise candidate generation. Therefore, algorithms for HUIs mining are generally slower than FIM algorithms. Two-phase based algorithms consume a lot of time to perform join operations. Therefore, these algorithms are not efficient to mine HUIs. They suffer from two main problems; firstly multiple scans of the dataset and secondly they generate too many candidate itemsets.

[1] All supersets of an infrequent itemset are infrequent and all subsets of a frequent itemset are frequent.

To overcome these limitations, tree-based algorithms such as IHUP [11], HUC-Prune [23], UP-Growth [12] and UP-Growth+ [21] have been proposed in literature to mine HUIs without expensive candidate generation and test. Tree-based algorithms also generate lot of candidate itemsets. To overcome the problem of candidate generation, single-phase algorithms such as HUI-Miner [24], HUP-Miner [25], d2HUP [26], FHM [27] and EFIM [28] came into limelight. HUI-Miner introduced a new data structure to store the information of itemsets named utility-list, which calculates the utility of itemsets without scanning the dataset and prune the search space efficiently. HUP-Miner, d2HUP and FHM also use the same utility-list based structure. Several utility mining techniques have been implemented for various practical applications.

The objective of this dissertation is to develop techniques and algorithms for mining HUIs from transactional datasets. We also work on how to efficiently identify the most significant HUIs. This thesis includes five works for mining HUIs. The issues and challenges of each work are discussed below.

1.2.1 Constraint-based High Utility Itemsets Mining

Although HUIs mining uncover thousands of HUIs but the end user is particularly interested in only a long and more actionable itemsets. Efficient mining for only the itemsets that satisfy user-specified constraints is called constraint-based mining. Traditional HUIs mining algorithms discover a large number of itemsets which reduce the efficiency and effectiveness. In order to decrease the number of HUIs by removing very small and very long itemsets. Constraint-based HUIs mining play an important role for decreasing the number of HUIs and finding more relevant rules. However, developing efficient algorithm for mining constrained based HUIs is a challenging task. It poses four major challenges to mine HUIs mining with length constraints.

- First, the itemsets in HUIs mining neither support to monotonic property nor anti-monotonic property. FIM techniques with length constraints rely on anti-monotonicity property. Hence, FIM techniques cannot be directly applied to HUIs mining with length constraints .

- Second, HUIs mining algorithms consume more time and memory compare to FIM algorithms. Hence, a compact dataset storage structure and an efficient pruning strategy are required.
- Third, incorporation of length constraints with HUIs mining and overestimate the upper bounds with length constraints are also the major challenges.
- Fourth, how to incorporate the length constraints (min_length and max_length) with min_util threshold.

1.2.2 Top-k High Utility Itemsets Mining

High utility itemsets mining algorithms discover all the itemsets which satisfy a given minimum utility threshold. However, it is difficult for users to set a proper minimum utility threshold. A very small threshold produces a huge number of HUIs, whereas, a higher threshold produces a few itemsets. Therefore, specify minimum utility is difficult and time consuming process. In order to address this issue top-k based HUIs mining has been proposed where k is the number of itemsets to be found. However, developing an efficient top-k HUIs mining algorithm is not an easy task. The major challenges are listed below.

- Top-k HUIs mining algorithms consume more time and memory compared to simple HUIs mining algorithms. Hence, a compact dataset storage structure or dataset cost reduction technique is required.
- min_util is not given in advance; the algorithm needs to start the from 0 or 1 min_util. Hence, the challenge is how to raise min_util automatically without missing any top-k HUIs.
- Pruning search space is also a big challenge to reduce the candidate itemset when min_util is always set to 0 or 1. Hence, an efficient pruning strategy is also required.

1.2.3 High Utility Itemsets Mining with Negative Utility Value

Most of HUIs mining algorithms work only with positive utility values. However, in real-world, items are found with both positive and negative utility value. In traditional HUIs algorithm, when negative utility value is considered , the discovered itemsets cannot be complete itemsets. In literature, only few works [29, 30] have been proposed for mining with negative utility value. There are four major challenges to develop efficient algorithm for negative utility value.

- Develop an efficient algorithm for mining HUIs with negative utility.
- HUIs mining algorithms consume more time and memory compare to FIM algorithms. Hence, a compact dataset storage structure and an efficient pruning strategy are required.
- Requirement of a special data structure that allows computing the utility of HUIs in memory without producing candidates.
- Lastly, the resulting algorithm is required to be more efficient than the state-of-the-art algorithms.

1.2.4 Constraint-based High Utility Itemsets Mining with Negative Utility Value

In literature, all HUIs mining with negative utility algorithms generates lots of itemsets which include many very small and too long itemsets that have no or small significance in real-life situations. In order to remove and too long itemsets, constraint-based mining can be proposed. Constraint-based HUIs mining with negative utility has many challenges some of them are discussed below.

- Most of HUIs algorithms mine rules with positive utility value. Traditional HUIs algorithms lose some candidate itemsets while mining HUIs with negative item.

- The utility-list based algorithms consider the candidates which may not appear in the datasets which are not efficient. And they also mine lots of tiny itemsets that are not actionable.
- HUIs mining algorithms scan dataset more than once and mining with negative utility items is a very computationally expensive task. Hence, dataset scanning cost reduction techniques are needed.
- Summation based overestimation utility counting technique are not up-to the mark. Hence, length of itemsets become longer which creates problem to analyze the result itemsets. Hence, few number of more meaningful itemsets are required.

1.2.5 Closed High Utility Itemsets Mining with Negative Utility Value

Traditional HUIs mining algorithms generate lots of redundant itemsets. In order to overcome this limitation closed HUIs mining has been proposed which avoid redundant itemsets. In literature, all closed HUIs mining algorithms work only with positive utility value. Although, negative utility is commonly seen in real-world applications. There exist several challenges to develop efficient closed HUIs mining algorithm with negative value. The key challenges are discussed below.

- Most of HUIs algorithms mine rules with positive utility value. Traditional CHUIs algorithms lose some candidate itemsets while mining with negative item.
- The state-of-the-art algorithms consider those candidates which may not appear in the datasets. Therefore, these algorithms are not efficient.
- CHUIs mining algorithms scan dataset more than once and mining with negative utility items becomes computationally expensive. Hence, we need dataset scanning cost reduction techniques.
- To design overestimation strategies those follows the downward closure property for CHUIs.

1.3 Contributions

The primary goal in this thesis is to propose solutions of challenges of the existing HUIs mining approaches. For this, we developed two solutions. First, we utilize pattern-growth based data structures, a tight upper bound for prune the search space. Second, we utilize dataset projection and transaction merging based preprocessing techniques to reduce the dataset scanning cost. Furthermore, to improve the mining process, we utilize transaction merging and dataset projection based transaction merging techniques as a preprocessing. The contributions of the thesis, therefore, can be grouped into five parts, where each part corresponds to one of the proposed solutions.

1.3.1 Constraint-based High Utility Itemsets Mining

To address the challenges of constraint-based HUIs mining, we propose an algorithm named EHIL (**E**fficient **H**igh utility **I**temsets with **L**ength constraints). Our contributions are summarized as follows.

- The proposed algorithm utilizes dataset projection and transaction merging techniques which reduce dataset size and speed up the mining process.
- Two prune strategies are utilized to prune a large number of unpromising candidates. These pruning strategies are calculated using depth first search.
- An efficient array-based utility counting technique is employed to compute the upper-bounds. The array-based utility counting technique calculates the utility of itemsets without scanning the original dataset.
- We demonstrate the usefulness of the proposed techniques through rigorous experiments. The results show that proposed algorithm outperforms the state-of-the-art FHM+ algorithm in term of execution time and memory usages.

1.3.2 Top-k High Utility Itemsets Mining

To address the challenges of top-k HUIs mining, an efficient top-k HUIs mining algorithm named TKEH is proposed. The key contributions of this work are as follows:

- We utilize transaction merging and dataset projection techniques to reduce the dataset scanning cost. These techniques reduce the dataset as larger items are explored.
- We employ three min_util threshold strategies for raising min_util automatically and efficiently.
- We utilize $EUCP$ and sup strategies to prune the search space efficiently.
- We utilize an efficient technique, that is utility array (UA), to calculate the utility of items and upper-bounds in linear time.

1.3.3 High Utility Itemsets Mining with Negative Utility Value

To address the challenges of HUIs mining with negative utility value, an efficient HUIs mining algorithm by considering both positive and negative utility is needed. We propose a novel algorithm, called EHIN (**E**fficient **H**igh-utility **I**temsets Mining with **N**egative Utility). It introduces several new ideas to discover HUIs efficiently with negative utility. The main contributions of this work are as follows:

- The proposed algorithm uses pattern growth approach and utilizes the dataset projection and merging techniques to reduce the memory requirement and speed up the execution time for mining process. Transaction merging is performed on both times before and after projection of the dataset. These techniques reduce the search space and enhance the execution.
- As previously mentioned, a key challenge in HUIs mining is how to prune the unpromising candidate itemsets, to reduce the search space. To address this challenge, two prune strategies redefined sub-tree utility and redefined local utility

are utilized to prune a large number of unpromising candidates. These pruning strategies are calculated using depth first search.

- To speed up the utility counting process, an efficient array-based utility counting technique is utilized to compute the redefined sub-tree utility and redefined local utility which allows to efficiently calculate the utility of itemsets without scanning the original dataset.
- The extensive experimental evaluation is carried on both real-life and synthetic benchmark datasets to evaluate the proposed algorithm. Results show that EHIN outperform the state-of-the-art FHN algorithm in term of execution time and memory usages.

1.3.4 Constraint-based High Utility Itemsets Mining with Negative Utility Value

To address the challenges of constraint-based HUIs mining with negative utility value, we propose an efficient algorithm named EHNL (**E**fficient **H**igh utility itemsets mining algorithm with **N**egative utility and **L**ength constraints). The key contributions of this work are summarized as follows:

- We propose an efficient algorithm for mining HUIs with negative utility items using a pattern-growth approach which only considers itemsets appeared in the dataset.
- We introduce minimum length constraint to remove the numerous tiny itemsets. We also use maximum length constraint to restrict the too longer itemsets.
- In order to reduce the dataset scanning cost, we utilize dataset projection and transaction merging techniques. A memory efficient array-based utility counting technique is also utilized to speed up the utility counting process.
- In order to prune the search space and to speedup the mining process we modify sub-tree based pruning strategy that was proposed by [28].

1.3.5 Closed High Utility Itemsets Mining with Negative Utility Value

To address the challenges of closed HUIs mining with negative utility value, we propose an efficient algorithm named CHN (**C**losed **H**igh utility itemsets mining algorithm with **N**egative utility). The key contributions of this work are summarized as follows:

- We propose an efficient algorithm for mining CHUIs with negative utility items using a pattern-growth approach by considering itemsets appeared in the dataset.
- In order to reduce the dataset scanning cost, we utilize dataset projection and transaction merging techniques.
- An efficient array-based utility counting technique is also utilized to speed up the utility counting process.
- In order to prune the search space and to speed up the mining process, we modify sub-tree based pruning strategy that was proposed by [28].
- We utilize a strict depth-first search order and can output the CHUIs. Also, we utilize bi-directional extension technique to check closure and prune the search space.
- We develop two versions of CHN named CHN(RSU-Prune) and CHN(TM). Extensive experimental results are evaluated to check the influence of the design techniques for CHN.

1.4 Organization

This thesis work presents five solutions to HUIs mining from transactional datasets as shown in FIGURE 1.1. First two works (EHIL & TKEH) presents solutions to positive utility-based mining. Rest three (EHIN, EHNL & CHN) works presents solutions to both positive and negative utility-based mining. FIGURE 1.1 shows the hierarchical structure of this thesis. The chapter-wise organizatoin of this thesis is shown in FIGURE 1.2. The thesis is organized into eight chapters. The chapter-wise organization of this thesis is as follows.

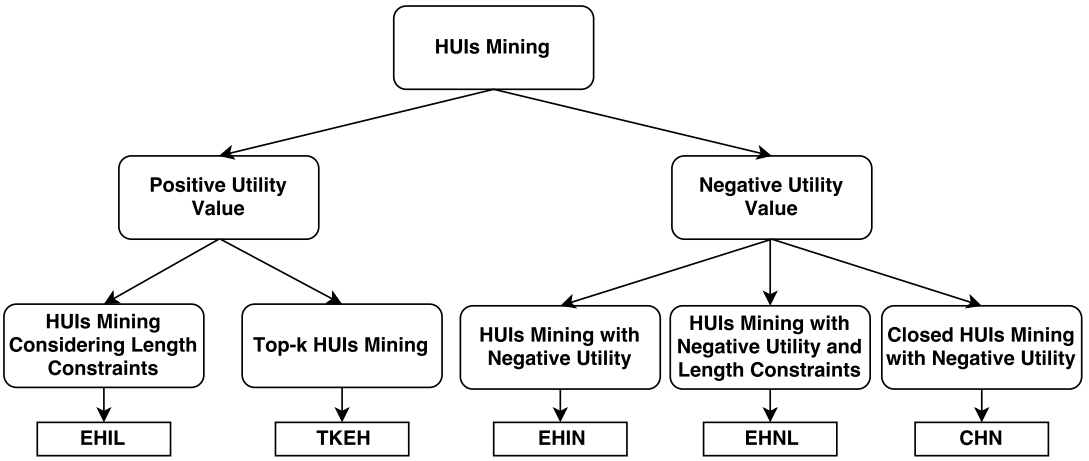

FIGURE 1.1: Structure of Thesis with Proposed Algorithms

Chapter 2 provides a systematic literature review and analysis of the state-of-the-art in HUIs mining. This literature review describes existing work in HUIs mining, constraint-based HUIs mining, top-k HUIs mining, closed HUIs mining and HUIs mining with negative utility value. Chapter 2 also presents the definitions of the problems that are explored in this thesis work.

Chapter 3 presents constraint-based HUIs mining. This chapter focuses on the length constraints such as maximum length and minimum length. An efficient algorithm EHIL (**E**fficient **H**igh utility **I**temsets with **L**ength constraints) is designed to identify HUIs.

Chapter 4 presents a top-k HUIs mining algorithm named TKEH. TKEH utilizes three strategies to raise internal minimum utility. The experiments are conducted on both synthetic and real datasets. The results show that TKEH incorporating the efficiency-enhanced strategies demonstrates impressive performance without missing any HUIs.

Chapter 5 presents a HUIs with negative utility value mining algorithm name EHIN (**E**fficient **H**igh-utility **I**temsets Mining with **N**egative Utility). Tree-based upper bounds are proposed to prune candidate itemsets.

Chapter 6 presents a HUIs with negative utility and length constraints based algorithm named EHNL (**E**fficient **H**igh utility itemsets mining algorithm with **N**egative utility and **L**ength constraints). Chapter 6 incorporate length constraints on HUIs with negative utility mining.

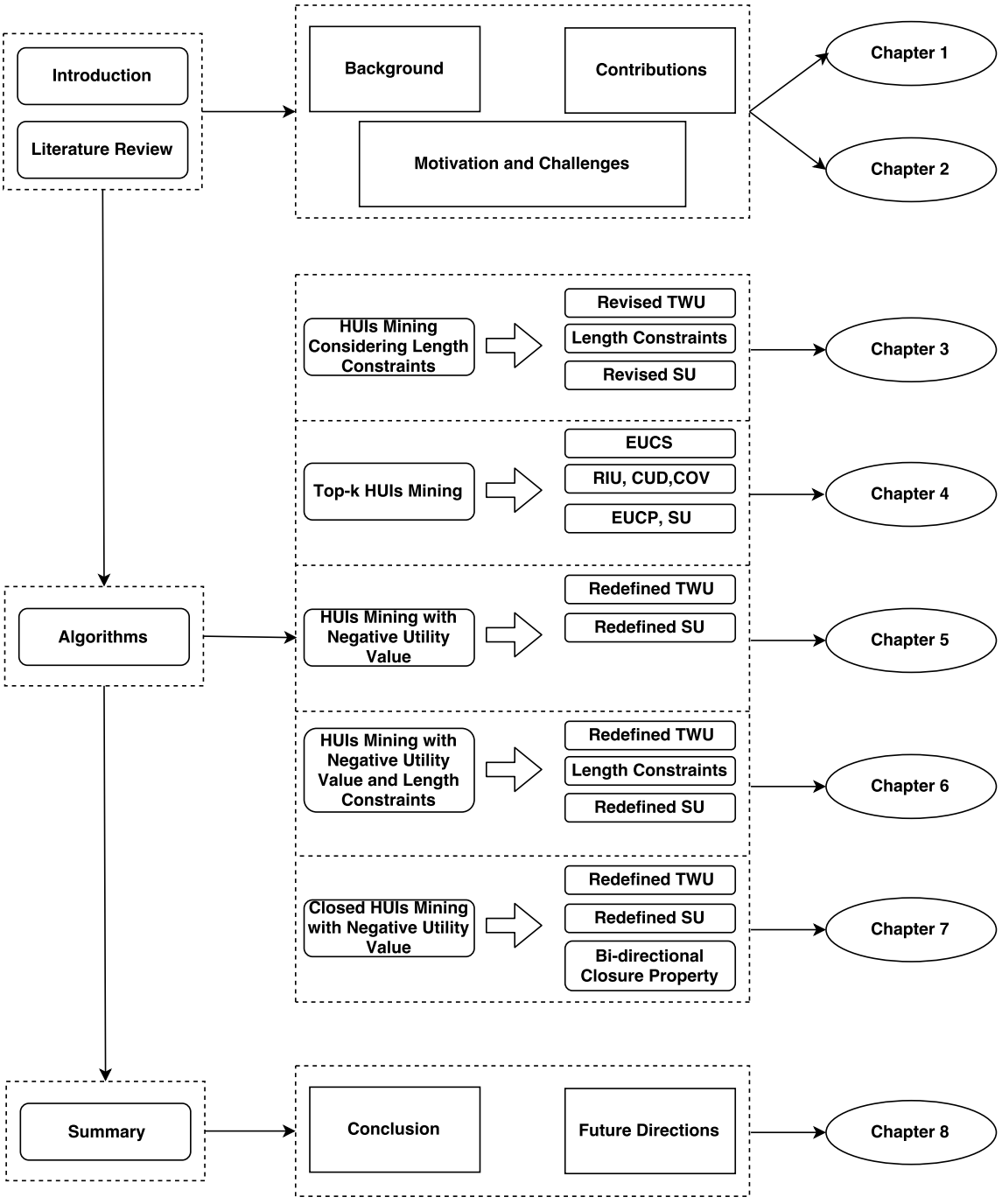

FIGURE 1.2: Organization of the Thesis

Chapter 7 proposes a novel concise algorithm to discover closed HUIs. An efficient algorithm named CHN is introduced to extract closed HUIs mining. Bi-directional closure based strategies are used to enhance the performance of CHN. Both real and synthetic datasets are used in our empirical studies. The results show that the proposed

approach is very efficient for a massive reduction in the number of HUIs without the loss of information and leads to better performance.

Chapter 8 concludes this thesis by providing a brief summary of what has been done and also provide the directions for future.