Big Data Research ••• (••••) •••–•••

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

Approximate Parallel High Utility Itemset Mining

Yan Chen *, Aijun An

Department of Electrical Engineering and Computer Science, York University, Toronto, Ontario, Canada

ARTICLE INFO

Article history:
Received 29 January 2016
Received in revised form 17 June 2016
Accepted 24 July 2016
Available online xxxx

Keywords:
Approximate
Parallel
Hadoop
Spark
HUI

ABSTRACT

High utility itemset mining discovers itemsets whose utility is above a given threshold, where the utility measures the importance of an itemset. It overcomes the limitation of frequent pattern mining, which uses frequency as its quality measure. To speed up the performance for mining high utility itemsets, many algorithms have been proposed which usually focus on optimizing the candidate generation process. However, memory and time performance limitations still cause scalability issues, especially when the dataset is very large.

In this paper, the problem is addressed by proposing a distributed parallel algorithm, *PHUI-Miner*, and a sampling strategy, which can be used either separately or simultaneously.

PHUI-Miner parallelizes the state-of-the-art high utility itemset mining algorithm *HUI-Miner*. In *PHUI-Miner*, the search space of the high utility itemset mining problem is divided and assigned to nodes in a cluster, which splits the workload. The sampling strategy investigates the required sample size of a dataset, in order to achieve a given accuracy. The sample size is selected based on a new theorem, which provides a theoretical guarantee on the accuracy of results. We also propose an approach combining sampling with PHUI-Miner, which mines an approximate set of results, but could provide better time performance.

In our experiments, we show that *PHUI-Miner* has high performance on different datasets and outperforms the state-of-the-art non-parallel algorithm *HUI-Miner*. The sampling strategy achieves accuracies much higher than the guarantee provided by the theorems in practice. Extensive experiments are also conducted to compare the time performance of PHUI-Miner with and without sampling.

1. Introduction

Frequent pattern mining has been an important topic since the concept of frequent itemsets was first introduced by Agrawal et al. [1]. Given a dataset of transactions, frequent pattern mining finds the itemsets whose support (i.e. the percentage of transactions containing the itemset) is no less than a given minimum support threshold. However, neither the number of occurrences of an item in a transaction, nor the importance of an item, is considered in frequent pattern mining. Itemsets with more occurrences or importance may be more interesting to users, since they may bring more profit.

In light of this, high utility itemset mining has been studied [2–5]. In high utility itemset mining, the term utility refers to the importance of an itemset; e.g., the total profit the itemset brings. An itemset is a *High Utility Itemset* (HUI) if the utility of the itemset is no less than a given minimum threshold. High utility itemset mining focuses more on the utility values in the dataset, which are usually related to profits for the business. Such utilities are interesting to the business owners, who could gain more profits from them. For example, supermarkets use frequent itemset mining to find merchandises customers usually buy together, so as to make recommendations to customers. However, with high utility itemset mining, supermarkets will be able to recommend not only the merchandises people usually buy together, but also the merchandises which will lead to more profits for the store.[1]

Most of the frequent pattern mining algorithms prune off itemsets in an early stage based on the popular *Apriori* property [6]: every sub-pattern of a frequent pattern must be frequent (also called the *downward closure property*). However, this property does not hold in high utility itemset mining, which makes mining high utility itemsets more challenging. The state-of-the-art approaches achieve good performance when the dataset is relatively small. However, the volume of data can grow so faster than expected,

* Corresponding author.
E-mail addresses: ychen@cse.yorku.ca (Y. Chen), aan@cse.yorku.ca (A. An).

[1] In Section 6.3, another example will be given to show a real world application of high utility itemset mining for news recommendation.

http://dx.doi.org/10.1016/j.bdr.2016.07.001
2214-5796/

that a single machine may not be able to handle a very large amount of data.

One option to solve the problem of large volumes of data is to use parallel distributed computing techniques. The MapReduce framework [7] (e.g., Hadoop) has been a popular solution recently, which enables scalable and fault-tolerant distributed processing of huge data on large clusters. Applications in the MapReduce framework have to conform the protocols of *mapper* and *reducer* as a disk-based paradigm, which restricts the flexibility as well as the performance of the algorithm. Spark is also a distributed computing framework, which is memory-based, and thus provides performance up to 100 times faster than Hadoop for certain applications [8]. Spark uses Resilient Distributed Dataset (RDD), which is a distributed memory abstraction, for in-memory computation of data, allowing efficient reuse of data.

For very large datasets, obtaining exact results is sometimes infeasible. Recent studies focus on mining an approximate set of frequent itemsets. In most cases, approximate solutions may already be satisfactory to users. In general, approximation methods can be divided into two categories: *pattern compressing* [9–12] and *sampling* [13–16]. Sampling is a method that mines approximate results from a sample of the entire dataset. The most important step in sampling is to decide the size of the sample we need in order to obtain a certain accuracy, which is also the focus of our sampling strategy proposed in this paper.

In this paper, we address the problem of high utility itemset mining by proposing *PHUI-Miner* (Parallel High Utility Itemset Miner) and a sampling strategy. *PHUI-Miner* is a parallel distributed algorithm, which parallelizes *HUI-Miner*, a state-of-the-art algorithm for high utility itemset mining. The sampling strategy provides the required sample size for a dataset in order to achieve a given accuracy.[2] It can be used together with any exact high utility itemset mining algorithm. To the best of our knowledge, this is the first piece of work to utilize sampling in high utility itemset mining. Our contributions are summarized as follows:

- *PHUI-Miner*, a parallel distributed algorithm, is proposed for parallel mining of high utility itemsets without sampling, which could lead to exact results.
- We propose and prove a new theorem, which shows the relationship between the high utility itemset mining results from the whole dataset, and those from a sample of it. The theorem leads to a sampling method with theoretical guarantees on the probability that an HUI can be returned and on the utility of a returned itemset. A feature of this sampling method is that the sample size required to achieve the theoretical guarantees is independent of the size of the original data, and is thus not necessarily going up as the data set grows.
- We also propose *PHUI-Miner with sampling*, an approach combining sampling with *PHUI-Miner*, which mines an approximate set of high utility itemsets, but achieves better performances.
- Extensive experiments are conducted and the time performance and scalability of *PHUI-Miner* are evaluated. *PHUI-Miner* is demonstrated to outperform the state-of-the-art non-parallel high utility itemset mining algorithm *HUI-Miner*. The time performance of *PHUI-Miner with sampling* is also evaluated, which is shown to be better than using *PHUI-Miner* alone. Furthermore, the accuracy of the sampling strategy is evaluated with several datasets and different parameters. Our results show that our sampling strategy achieves accuracy even higher than the expectations based on our theoretical analysis.

The paper is organized as follows. Section 2 is a literature survey of related work. Section 3 introduces relevant definitions and a problem statement. Section 4 presents *PHUI-Miner*. Section 5 describes the sampling strategy and *PHUI-Miner with sampling*. We show experimental results in Section 6, and conclude the paper in Section 7.

2. Related work

Before the problem of high utility itemset mining was first proposed by Yao et al. [17], a variation of the problem, named share frequent itemsets mining, was studied by many researchers. Several algorithms have been proposed: e.g., ZP [18], ZSP [18], FSH [19], ShFSH [19], and DCG [20]. These algorithms can be used to mine high utility itemsets. However, they all use the Apriori [21] like strategy, which results in the problem of repeated database scans and large numbers of candidates. To improve the performance of these algorithms, Liu et al. proposed Two-phase [22], which uses an important utility measure, named *Transaction Weighted Utility* (TWU), for pruning the search space, since the *downward closure property* is not applicable in high utility itemset mining. Afterwards, another pruning strategy, called the *isolated items discarding strategy* (IIDS), was proposed in FUM [23] and DCG+ [23]. The number of candidates are largely reduced by these pruning strategies. However, the problem of repeated database scans is still not solved. An algorithms based on FP-Growth algorithm [24] have been proposed to mine high utility itemsets with at most three scans of database, and thus have better performance. Examples of these algorithms include IHUP [2], HUC-Prune [25], UP-Growth [4], UP-Growth+ [26]. However, the candidate itemsets are still too many compared to the high utility itemsets. HUI-Miner [5] is one of the recent efficient algorithms proposed by Liu et al. demonstrated to have an order of magnitude better performance than other algorithms.

Parallel distributed algorithms solve the problem of mining massive datasets. Several studies [27–32] have been done for mining frequent patterns in distributed environments, inspired by the MapReduce framework proposed by Google [7]. Some of them [27–29] use a naive approach which computes the support of every itemset in the dataset in a single MapReduce round, resulting in huge data replication. An adaption of FP-Growth algorithm to MapReduce, called PFP [30], is a more sophisticated approach. Given a minimum frequency threshold, PFP first applies a parallel and distributed counting approach to compute the frequent items. The frequent items are then partitioned into groups randomly. Subsequently, the dataset is used to generate group-dependent transactions, which are sent to reducers. Finally, the reducers use an FP-Growth like approach to generate group-dependent frequent itemsets. However, very few studies [33] have been conducted on high utility itemset mining with distributed computing techniques so far. In [33], a parallelized version of UP-Growth based on MapReduce [7] is presented. Since MapReduce is known to be much slower than Spark [34] and UP-Growth is shown to be much slower than HUI-Miner [5], our proposed method that is based on Spark and HUI-Miner is much faster than the one in [33].

As the volume of data grows, the mining task consumes more and more time. Mannila et al. [35] first suggested that sampling can be used to efficiently obtain association rules. Then Toivonen [13] presented a sampling algorithm, which builds a complete set of association rules with a probability depending on the sample size. The Chernoff bound and the union bound are used, in which the Bernoulli random variable refers to whether an itemset appears in a transaction. A number of previous works [15,36–41] have been

[2] The accuracy of the sampling strategy is measured by the precision, recall or relative utility error (to be defined in Section 6.2) of the discovered patterns from the sample.