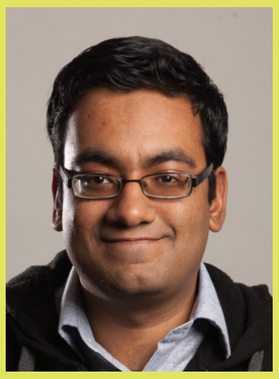
Ph.D. Thesis Defense of Siddharth Dawar

Title: Mining high-utility itemsets from a transaction database

Ph.D. Thesis Defense

Date: April 17, 2021, Time: 10:00 AM (IST)



Examiners:

- **Prof. P. Krishna Reddy, IIIT-H**
- **Prof. Vasudha Bhatnagar, University of Delhi**
- **Prof. Bac Le, HoD, Professor, HCMUS, Vietnam**

Advisors: Dr. Vikram Goyal, Dr. Debajyoti Bera

Abstract: Technological advances have enabled organizations to store large amounts of data cost-effectively. Patterns hidden in such massive databases can be valuable to industries to gain actionable knowledge and promote their business. For example, a retail store can utilize information about the products frequently purchased together by its customers for shelf-space management and inventory management. Frequent itemset mining has been studied extensively by the research community to mine such patterns from a transaction database. A transaction represents the set of products purchased together by a customer in the above-mentioned example. An example of a frequent itemset can be products like milk and bread purchased together frequently by customers from a retail store. Frequent itemset mining assumes that the items present in a transaction database have equal importance. However, customers purchase products in different quantities, and products generate different profits for the retail store. The notion of mining high-utility itemsets was formulated by researchers to mine such a set of items. For example, high-utility itemset mining can extract the set of profitable products purchased by customers from a retail store.

High-utility itemset mining is a generalization of the frequent itemset mining problem that associates a positive weight with each item in a transaction. The utility of an itemset in a transaction is the sum of the weights associated with its items. High-utility itemset mining is a more challenging problem compared to frequent itemset mining as the utility measure is neither monotone nor anti-monotone, unlike for frequent itemset mining. During a search space exploration, if the frequency of an itemset is less than the minimum frequency threshold, the itemset and its supersets can not be frequent. However, the superset of a low-utility itemset can have a high-utility, and the subset of a high-utility itemset can be a low-utility itemset. So, the search space can not be pruned solely based on the utility of a partially explored itemset.

In this thesis, we analyze, \enquote{how can we improve the performance of existing high-utility itemset mining algorithms?} The existing algorithms for mining high-utility itemsets can be categorized into one-phase and two-phase algorithms. The two-phase tree-based algorithms for mining high-utility itemsets generate candidate high-utility itemsets in the first phase by constructing a tree data structure recursively, and compute the utility of candidate itemsets through another database scan in the second phase called the verification phase. It has been observed by researchers that the performance of such two-phase tree-based algorithms can be improved by reducing the number of generated candidates. We begin by proposing a novel tree data structure called UP-Hist tree that augments a histogram of item weights frequency and a two-phase tree-based algorithm called UP-Hist Growth. We compare the performance of UP-Hist Growth against the state-of-the-art two-phase tree-based algorithms on several benchmark sparse and dense datasets. Our results demonstrate that the UP-Hist Growth algorithm performs better than those algorithms.

We observe in our experimental study that the two-phase tree-based algorithms, including UP-Hist Growth, run out of memory on some of the dense datasets and the state-of-the-art list-based algorithms like HUI-Miner perform faster than the two-phase tree-based algorithms on dense datasets. We also observe that the tree-based algorithms generate candidates quickly in the first phase, but spend a lot of time in the verification phase. The list-based algorithms construct an inverted-list data structure for every itemset by intersecting the inverted-lists of its immediate subsets. The intersection operation can become a performance bottleneck for list-based algorithms, and list-based algorithms can also generate itemsets that are non-existent in the database during search-space exploration. To combat the limitations of these approaches, we propose a hybrid algorithm that can harness the benefits of both types of algorithms by combining any tree-based algorithm with a list-based algorithm to extract high-utility itemsets. As a case study, we construct two hybrid algorithms by joining two tree-based algorithms named UP-Hist Growth and UP-Growth+ with a list-based algorithm called FHM. Our experimental study validates that the hybrid algorithms have less total execution time compared to the existing two-phase tree-based algorithms on sparse and dense datasets. Additionally, the hybrid algorithms also perform better than the list-based algorithms on sparse datasets.

We observe that a faster high-utility itemset mining algorithm can be designed by augmenting the inverted-list data structure on the top of a tree data structure as it can reduce the amount of information stored within the tree and also cost of the intersection operation during the search-space exploration. To implement this idea, we propose a tree structure called UT$_$Mem-tree that augments information compactly in a HashMap with each node of the tree and design the first \enquote{one-phase tree-based} algorithm called UT-Miner to mine high-utility itemsets. We also propose a mechanism to construct a lightweight projected database during the mining process for superior performance, especially for dense datasets. We also conduct experiments to compare the performance of UT-Miner with the state-of-the-art high-utility itemset mining algorithms. The results confirm that UT-Miner performs better than the state-of-the-art tree-based, list-based, and hybrid algorithms on sparse and dense datasets.

The existing algorithms and data structures for mining high-utility itemsets are designed for a specific utility function only. We explore the possibility of designing data structures and algorithms that can mine high-utility itemsets for any subadditive monotone utility function. In this scenario, the utility of an itemset in a transaction need not be the sum of its item utilities. We design tighter upper-bounds and algorithms that can mine high-utility itemsets for such utility functions. We believe that generalization of utility functions can be useful. To demonstrate this we identify an application of high subadditive-monotone utility itemsets to find active high-influential groups of users from a Twitter dataset for applications like viral marketing.

In this thesis, we have explored how to improve the state-of-the-art techniques in high-utility itemset mining. We designed tighter bounds to reduce the search space exploration and algorithms for a class of utility functions that can generalize the classical addition function used by the existing algorithms.