Max-FTP: Mining Maximal Fault-Tolerant Frequent Patterns from Databases

Shariq Bashir and A. Rauf Baig

National University of Computer and Emerging Sciences, Islamabad, Pakistan
shariq.bashir@nu.edu.pk, rauf.baig@nu.edu.pk

Abstract. Mining Fault-Tolerant (FT) Frequent Patterns in real world (dirty) databases is considered to be a fruitful direction for future data mining research. In last couple of years a number of different algorithms have been proposed on the basis of Apriori-FT frequent pattern mining concept. The main limitation of these existing FT frequent pattern mining algorithms is that, they try to find all FT frequent patterns without considering only useful long (maximal) patterns. This not only increases the processing time of mining process but also generates too many redundant short FT frequent patterns that are un-useful. In this paper we present a novel concept of mining only maximal (long) useful FT frequent patterns. For mining such patterns algorithm we introduce a novel depth first search algorithm **Max-FTP** (Maximal Fault-Tolerant Frequent Pattern Mining), with its various search space pruning and fast frequency counting techniques. Our different extensive experimental result on benchmark datasets show that Max-FTP is very efficient in filtering un-interesting FT patterns and execution as compared to Apriori-FT.

Keywords: Fault Tolerant Frequent Patterns Mining, Maximal Frequent Patterns Mining, Bit-vector Representation, and Association Rules.

1 Introduction

Mining frequent patterns from transactional or relational datasets with support greater than a certain user defined threshold, plays an important role in many data mining applications such as intrusion detection, finding gene expression patterns, web log patterns etc. In recent years, a number of algorithms have been proposed for efficient mining of such frequent patterns, on the basis of Apriori property proposed by Agrawal et al. [1]. These algorithms take a transactional dataset and support threshold *(min_sup)* as an input and output those exact matching frequent patterns which contain support greater than *min_sup*, with assuming that the dataset is very well pre-processed and noise free. However, the real world datasets are dirty and contain missing and noisy values. In such situations, users face difficulties in setting this *min-sup* threshold to obtain their desired results. If *min-sup* is set too large, then there may be a small number of frequent patterns, which does not give any desirable result. If the *min-sup* is set too small, then there may be many redundant short un-useful frequent patterns, which not only take a large processing time for mining but also increase the

R. Cooper and J. Kennedy (Eds.): BNCOD 2007, LNCS 4587, pp. 235–246, 2007.

complexity of filtering un-interesting frequent patterns. In both situations, the ultimate goal of mining interesting frequent patterns is undermined.

For handling such situations, J. Pei at el. in [6] introduced a new application of finding only interesting frequent patterns in a real world dirty datasets, instead of finding exact patterns. This approach is known as fault-tolerant (FT) frequent pattern mining. The problem of mining all FT frequent patterns from a dirty transactional dataset can be considered from the following two conditions [6].

1. Under user defined fault tolerance factor δ, a pattern X with cardinality greater than δ is called a FT frequent pattern, if it appears in at least k number of FT-transactions. A transaction t is called a FT-transaction under fault tolerance factor δ, if it contain at least $|X|$-δ number of items of X. The number k is called the frequency of X which must be greater or equal than the minimum FT support threshold (min_sup^{FT}).
2. Each individual single item i of X must be appeared in at least l number of FT-transaction of X, where l is called the minimum item support threshold under fault tolerance factor δ ($\text{item\_sup}^{FT}{}_{\delta}$).

For example, with $\text{min\_sup}^{FT} = 3$ and $\text{item\_sup}^{FT}{}_{\delta} = 2$, the pattern $\langle A, B, C, D \rangle$ is a FT frequent pattern under fault tolerance factor $\delta = 1$, since 3 out of 4 items are present in FT-transaction T1, T3 and T5 which qualifies min_sup^{FT} threshold and each single item A, B, C and D is present in at least 2 transactions with qualifies $\text{item\_sup}^{FT}{}_{\delta}$ threshold. In [6] they also proposed an Apriori-FT algorithm for finding all type of such patterns. The Apriori-FT was extended from the Apriori approach, in which downward closure property is used for mining FT frequent patterns. Similar to Apriori algorithm, Apriori-FT applies a bottom-up search that enumerates every single FT frequent pattern. This implies that in order to produce a FT frequent pattern of length l, it must produce all 2^l of its subsets, since they too must be frequent FT. This exponential complexity fundamentally restricts Apriori-FT like algorithms in discovering only useful interesting FT frequent patterns in a reasonable time limit. Moreover, mining FT frequent patterns are very complex than mining all frequent patterns, in terms of both search space exploration and frequency counting of candidate patterns. In frequent pattern mining, a candidate pattern X is declared to be frequent, by checking its frequency in only one dataset scan. While in FT frequent pattern mining, a number of dataset scans are needed to declare a candidate FT pattern X as frequent, which depends on the cardinality of pattern X. In addition to frequency counting, most of the search space pruning techniques, such as parent equivalence pruning (PEP) and 2-Itemset Pair of frequent pattern mining can not be applied on mining FT frequent patterns for filtering infrequent FT patterns.

To overcome these limitations, in this paper we have introduced a novel maximal or long FT frequent pattern mining (MFP^{FT}) concept. Similar to maximal frequent pattern mining [3], a pattern X is called a maximal FT frequent pattern, if it has no superset that is also a maximal frequent FT pattern. Mining only MFP^{FT}s has many advantages over mining all FT frequent patterns. Firstly, long patterns are very useful in some very important data mining applications such as biological data from the field of DNA and protein analysis and clustering. Secondly, different search space pruning techniques such as FHUT and HUTMFI (Section 5) can be also applied easily on