Partition-Based Approach to Processing Batches of Frequent Itemset Queries

Przemyslaw Grudzinski, Marek Wojciechowski, and Maciej Zakrzewicz

Poznan University of Technology
Institute of Computing Science
ul. Piotrowo 2, 60-965 Poznan, Poland
{marek, mzakrz}@cs.put.poznan.pl

Abstract. We consider the problem of optimizing processing of batches of frequent itemset queries. The problem is a particular case of multiple-query optimization, where the goal is to minimize the total execution time of the set of queries. We propose an algorithm that is a combination of the Mine Merge method, previously proposed for processing of batches of frequent itemset queries, and the Partition algorithm for memory-based frequent itemset mining. The experiments show that the novel approach outperforms the original Mine Merge and sequential processing in majority of cases.

1 Introduction

Discovery of frequent itemsets [1] is a very important data mining problem with numerous practical applications. Informally, frequent itemsets are subsets frequently occurring in a collection of sets of items. Frequent itemsets are typically used to generate association rules. However, since generation of rules is a rather straightforward task, the focus of researchers has been mostly on optimizing the frequent itemset discovery step.

Frequent itemset mining (and in general, frequent pattern mining) is often regarded as advanced querying where a user specifies the source dataset, the minimum support threshold, and optionally pattern constraints within a given constraint model [9]. A significant amount of research on efficient processing of frequent itemset queries has been done in recent years, focusing mainly on constraint handling and reusing results of previous queries [5][7][12][13].

Recently, a new problem of optimizing processing of batches of frequent itemset queries has been considered [19][20]. The problem was motivated by data mining systems working in a batch mode or periodically refreshed data warehouses, but is also relevant in the context of multi-user, interactive data mining environments. It is a particular case of multiple-query optimization [18], well-studied in database systems. The goal is to find an optimal global execution plan, exploiting similarities between the queries.

So far, two methods of processing batches of frequent itemset queries have been proposed: Mine Merge [19] and Common Counting [20]. Both methods exploit the overlapping between queries' datasets to reduce the overall processing time. Unfortunately, both methods have serious limitations, which is the motivation for further research on the topic.

H. Legind Larsen et al. (Eds.): FQAS 2006, LNAI 4027, pp. 479 – 488, 2006.

Common Counting consists in concurrent executing of a frequent itemset mining algorithm for the queries, and integrating dataset scans performed by the queries. Common Counting was designed to work with Apriori [3], in case of which it needs to maintain candidate hash-trees of several queries in main memory. If not all the hash-trees fit into memory, the queries have to be scheduled into phases, which degrades Common Counting's performance. Application of Common Counting to newer pattern-growth mining algorithms [8] is problematic as these algorithms store a compressed form of the database in main memory, which may be infeasible for more than one query at the same time, even for today's machines.

The idea of Mine Merge is to transform the original batch of overlapping queries into the set of intermediate non-overlapping queries operating on dataset partitions, whose boundaries are defined by the overlapping between the original queries. After executing the intermediate queries, the answers to original queries are generated using the method proposed in [17] for memory-based partitioning. Mine Merge is not bound to a particular mining algorithm and its memory requirements are not greater that those of the basic mining algorithm applied to intermediate queries. The disadvantage of Mine Merge is that it requires significant overlapping between the queries in order to compensate the extra database scan needed to consolidate the results from intermediate queries.

In this paper we propose a novel method for processing batches of frequent itemset queries, called PMM+ (Partition Mine Merge Improved), which combines disk-based partitioning of Mine Merge with memory-based partitioning of the well-known Partition algorithm from [17]. The advantage of the new method is that it requires exactly two scans of the union of source datasets of the queries forming a batch.

The paper is organized as follows. In Section 2 we review related work. Section 3 contains basic definitions regarding frequent itemset queries and reviews the Mine Merge method for processing of batches of frequent itemset queries. The motivations underlying PMM+ and the new method itself are presented in Section 4. In Section 5 we present and discuss results of experiments conducted to evaluate performance of PMM+. Section 6 contains conclusions.

2 Related Work

Multiple-query optimization has been extensively studied in the context of database systems (see [18] for an overview). The idea was to identify common subexpressions and construct a global execution plan minimizing the overall processing time by executing the common subexpressions only once for the set of queries [4][10][15]. Data mining queries could also benefit from this general strategy, however, due to their different nature they require novel multiple-query processing methods.

To the best of our knowledge, the only two multiple-query processing methods for data mining queries are Mine Merge [19] and Common Counting [20], mentioned above. Recently, the need for multiple-query optimization has been postulated in the somewhat related research area of inductive logic programming, where a technique based on similar ideas as Common Counting has been proposed, consisting in combining similar queries into query packs [6].