Chapter 10
(Approximate) Frequent Item Set Mining Made Simple with a Split and Merge Algorithm

Christian Borgelt
European Center for Soft Computing, Spain

Xiaomeng Wang
Otto-von-Guericke University of Magdeburg, Germany

ABSTRACT
In this chapter the authors introduce SaM, a split and merge algorithm for frequent item set mining. Its core advantages are its extremely simple data structure and processing scheme, which not only make it very easy to implement, but also fairly easy to execute on external storage, thus rendering it a highly useful method if the data to mine cannot be loaded into main memory. Furthermore, the authors present extensions of this algorithm, which allow for approximate or “fuzzy” frequent item set mining in the sense that missing items can be inserted into transactions with a user-specified penalty. Finally, they present experiments comparing their new method with classical frequent item set mining algorithms (like Apriori, Eclat and FP-growth) and with the approximate frequent item set mining version of RElim (an algorithm the authors proposed in an earlier paper and improved in the meantime).

INTRODUCTION
It may not even be an exaggeration to say that the tasks of frequent item set mining and association rule induction started the popular research area of data mining. At least, however, these tasks have a strong and long-standing tradition in data mining and knowledge discovery in databases and account for a huge number of publications in data mining conferences and journals. The enormous research efforts devoted to these tasks have led to a variety of sophisticated and efficient algorithms to find frequent item sets. Among the best-known are Apriori (Agrawal and Srikant 1994, Agrawal et al. 1996), Eclat (Zaki et al. 1997) and FP-growth (Han et al. 2000).

DOI: 10.4018/978-1-60566-858-1.ch010
Nevertheless, there is still room for improvement: while Eclat, which is the simplest of the mentioned algorithms, can be fairly slow on some data sets (compared to other algorithms), FP-growth, which is usually the fastest algorithm, employs a sophisticated data structure and requires to load the transaction data to mine into main memory. Hence a simpler processing scheme, which still maintains efficiency, is desirable. Other lines of improvement include filtering the found frequent item sets and association rules (see, for example, [Webb and Zhang 2005, Webb 2007]), identifying temporal changes in discovered patterns (see, for example, [Böttcher et al. 2005, Böttcher et al. 2007]), and discovering fault-tolerant or approximate frequent item sets (see, for example, [Cheng et al. 2001, Pei et al. 2001, Wang et al. 2005]).

In this paper we introduce SaM, a split and merge algorithm for frequent item set mining. Its core advantages are its extremely simple data structure and processing scheme, which not only make it very easy to implement, but also fairly easy to execute on external storage, thus rendering it a highly useful method if the data to mine cannot be loaded into main memory. Furthermore, we present extensions of this algorithm, which allow for approximate or “fuzzy” frequent item set mining in the sense that missing items can be inserted into transactions with a user-specified penalty. We developed this algorithm as a simplification of the (already very simple) RElim algorithm (Borgelt 2005b), which we improved in the meantime.

The rest of this paper is structured as follows: first we briefly review the fundamentals of frequent item set mining, and especially the basic divide-and-conquer scheme underlying many frequent item set mining algorithms. Secondly, we present our SaM (Split and Merge) algorithm for exact frequent item set mining and compare it experimentally to classic frequent item set mining algorithms like Apriori, Eclat, and FP-growth, but also our own RElim algorithm (Borgelt 2005b). In the next step we review approximate or “fuzzy” frequent item set mining in the sense that missing items can be inserted into transactions with a user-specified penalty. Based on this review we present two extensions of our SaM algorithm that allow to perform such approximate frequent item set mining with unlimited and limited item insertions, respectively. These extensions are then experimentally compared to the corresponding extensions of the RElim algorithm (Wang et al 2005). Finally, we draw conclusions from our discussion and experiments.

**Frequent item set mining**

Frequent item set mining is a data analysis method that was originally developed for market basket analysis. It aims at finding regularities in the shopping behavior of the customers of supermarkets, mail-order companies and online shops. In particular, it tries to identify sets of products that are frequently bought together. Once identified, such sets of associated products may be exploited to optimize the organization of the products on the shelves of a supermarket or on the pages of a mail-order catalog or web shop, may be used to suggest other products a customer could be interested in, or may give hints which products may conveniently be bundled.

Formally, the task of frequent item set mining can be described as follows: we are given a set $B$ of items, called the item base, and a database $T$ of transactions. Each item represents a product, and the item base represents the set of all products offered by a store. The term item set refers to any subset of the item base $B$. Each transaction is an item set and represents a set of products that has been bought by an actual customer. Since two or even more customers may have bought the exact same set of products,