Dynamic Itemset Hiding Algorithm for Multiple Sensitive Support Thresholds

Ahmet Cumhur Öztürk, İzmir Institute of Technology, İzmir, Turkey
Belgin Ergenç, İzmir Institute of Technology, İzmir, Turkey

ABSTRACT

This article describes how association rule mining is used for extracting relations between items in transactional databases and is beneficial for decision-making. However, association rule mining can pose a threat to the privacy of the knowledge when the data is shared without hiding the confidential association rules of the data owner. One of the ways hiding an association rule from the database is to conceal the itemsets (co-occurring items) from which the sensitive association rules are generated. These sensitive itemsets are sanitized by the itemset hiding processes. Most of the existing solutions consider single support thresholds and assume that the databases are static, which is not true in real life. In this article, the authors propose a novel itemset hiding algorithm designed for the dynamic database environment and consider multiple itemset support thresholds. Performance comparisons of the algorithm is done with two dynamic algorithms on six different databases. Findings show that their dynamic algorithm is more efficient in terms of execution time and information loss and guarantees to hide all sensitive itemsets.

KEYWORDS
Dynamic Itemset Hiding, Itemset Hiding, Multiple Sensitive Support Thresholds, Privacy Preserving Association Rule Mining

1. INTRODUCTION

Data mining is the process of extracting knowledge from data with the help of statistics, artificial intelligence, machine learning and database systems. Association rule mining is one of the data mining tasks. It was first proposed by (Agrawal and Srikant, 1994) and is used for discovering correlated items transactional databases. Association rule mining process has mainly two steps; the first step is called frequent itemset (co-occurring items) generation and the second step is called rule generation where meaningful rules are generated from the discovered frequent itemsets. The second step is straightforward and similar in all proposed algorithms; as a result, association rule mining algorithms focus on the first step which is computationally expensive. For this reason, terms association rule mining and itemset mining are used interchangeably.

Lately many organizations use itemset mining tasks for short or long-term planning and strategical decision making. In modern business, organizations also share data with each other or with third parties in order to provide extraction of knowledge for mutual benefit. Similarly, itemset mining tasks are applied on this shared data however this may pose security threat for strategical and sensitive information of data owners.

The importance of this threat is well explained with a scenario given in (Clifton and Marks, 1996). Suppose BigMart supermarket chain is negotiating with DedTrees Paper Company for selling their

DOI: 10.4018/IJDWM.2018040103

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
products, and DedTrees offers to reduce their price if BigMart agrees to share sales database. After BigMart agrees to share the sales database, DedTrees applies itemset mining task on this database and finds out that people who purchase skim milk also purchase Green paper. Dedtrees Company then runs a coupon marketing campaign that gives 50 cents off for each purchase a Dedtrees product. The campaign cuts heavily sales of Green paper and as a result Green paper has to increase prices because of the low sales amount. In the next negotiation with DedTrees, they are unwilling to reduce their prices because they reach their goal. As a result, the BigMart suffers serious losses to competitors. This scenario shows that before the data is shared with other parties, the database owner should take precautions to protect its strategic and sensitive knowledge from being discovered by itemset mining task. Privacy preserving itemset mining is the problem of preserving the sensitive itemsets from being discovered in case of data sharing.

The most popular approach for sanitizing sensitive and frequent itemsets is to decrease their frequency (support) under predefined support threshold. Such a modification operation of converting the original database D into a sanitized database D’ is called frequent itemset hiding. A well designed frequent itemset hiding algorithm should hide all given sensitive itemsets while keeping the loss of non-sensitive itemsets, production of new artificial frequent itemsets and the distortion done on the database minimum. Most proposed frequent itemset hiding approaches allow user to define a single support threshold for each sensitive itemset and assume that the databases are static (Amiri, 2007; Li et al., 2007; Weng et al., 2008; Dehkordi and Dehkordi, 2016; Verykios et al., 2004; Pontikakis et al., 2004; Hong et al., 2013; Cheng et al., 2016). Single support threshold barrier does not suit the nature of different itemsets; in transactional databases, frequency of some sensitive itemsets may be too high while some sensitive itemsets may be too low. Assigning unique single sensitive support threshold for each sensitive itemsets may result in decreasing the frequency of some itemsets more than required.

On the other hand, transactional databases are dynamic; they get updates continuously. When dynamicity of the databases is considered applying a sensitive itemset hiding algorithm from the start will result in redundant execution time and memory allocation. In (Dai and Chiang, 2010; Jadav et al., 2014) dynamic sanitization algorithms are proposed; they either do the sanitization on the whole database or on the update only. The proposed approach in (Dai and Chiang, 2010) uses a tree like data structure to speed up the execution time and does the sanitization on the whole database. This approach is good for minimizing the side effects and achieving optimum sanitization however the data structure used is inadequate for dense databases when the given set of sensitive itemsets contains a certain number of overlapping items. As the number of overlaps increase it becomes impossible to uncover all sensitive itemset supporting transactions from this data structure. These supporting transactions are called sensitive transactions and uncovering inadequate number of sensitive transactions may result in smaller search space of sensitive transactions. So the sanitized database may keep containing sensitive itemsets. The proposed approach in (Jadav et al., 2014) only modifies the transactions in the updated part and does not use any data structure to speed up the execution time. Although this approach guarantees hiding all sensitive itemsets, since the modification is always done on the incremental part, distortion on the database and loss of non-sensitive information cannot be kept at minimum.

In this paper a dynamic frequent itemset hiding algorithm DynamicPGBS with four major processes is proposed: Initialization, Increment Handling, Hiding and Publish Database. Hiding process is an extension of PGBS (Öztürk and Ergenç-Bostanoğlu, 2017). The DynamicPGBS is designed for hiding a given set of sensitive itemsets by deleting one or more items from adequate number of transactions while minimizing the execution time, memory requirement, distortion on the updated database and loss of non-sensitive knowledge. The main contributions of the dynamic algorithm are; 1) different sensitive support thresholds can be assigned to sensitive itemsets, 2) sanitization is done by considering whole transactions of the database that means large search space of sensitive transactions and less side effects on the sanitized database, 3) hiding process is designed for incremental environment, 3) it guarantees hiding all given sensitive itemsets. In the performance evaluation, DynamicPGBS is compared with similar counterparts SPITF (Dai and Chiang, 2010) and