Available online at www.sciencedirect.com
ScienceDirect
Information Sciences 177 (2007) 490–503
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

Privacy-preserving algorithms for distributed mining of frequent itemsets ☆

Sheng Zhong

Computer Science and Engineering Department, State University of New York at Buffalo, Amherst, NY 14260, USA

Received 11 October 2005; received in revised form 25 July 2006; accepted 7 August 2006

Abstract

Standard algorithms for association rule mining are based on identification of *frequent itemsets*. In this paper, we study how to maintain privacy in distributed mining of frequent itemsets. That is, we study how two (or more) parties can find frequent itemsets in a distributed database without revealing each party's portion of the data to the other. The existing solution for vertically partitioned data leaks a significant amount of information, while the existing solution for horizontally partitioned data only works for three parties or more. In this paper, we design algorithms for both vertically and horizontally partitioned data, with cryptographically strong privacy. We give two algorithms for vertically partitioned data; one of them reveals only the support count and the other reveals nothing. Both of them have computational overheads linear in the number of transactions. Our algorithm for horizontally partitioned data works for two parties and above and is more efficient than the existing solution.

Keywords: Data mining; Association rules; Distributed databases; Privacy

1. Introduction

Data mining has been studied extensively and applied widely [28,30,29]. Through the use of data-mining techniques, businesses can discover hidden patterns and rules in a database and then employ them to predict features of data items that have not yet arrived. An important scenario in data mining is *distributed data mining*, in which a database is distributed between two (or more) parties, and each party owns a portion of the data. These parties need to collaborate with each other so that they can jointly mine the data and produce results that are interesting to both of them. Privacy concerns are of great importance in this scenario, because each party may not want to reveal her own portion of the data, although she would like to participate in the mining.

☆ Work partly done while the author was at Yale University. A previous version of this paper appeared as Yale University Technical Report Yale-DCS-TR1255 in August 2003.

E-mail address: szhong@cse.buffalo.edu

0020-0255/$ - see front matter
doi:10.1016/j.ins.2006.08.010

This paper is concerned with a major category of data mining, namely mining of *association rules*. Consider the transaction database of a supermarket. We may find that most of those who buy bread also buy milk. Therefore, "bread ⇒ milk," which means "buying bread implies buying milk," is a candidate association rule. Two metrics are defined to measure such a candidate rule: *confidence* and *support*. Here confidence means the number of transactions in which both bread and milk are bought divided by the number of transactions in which bread is bought. Support means the number of transactions in which bread and milk are bought divided by the overall number of transactions. A candidate is considered a valid association rule if both its confidence and its support are sufficiently high.

Standard algorithms for association rule mining are based on identification of *frequent itemsets* [3]. We say that bread and milk constitute a frequent itemset if, in a sufficiently large percentage of transactions, both of them are bought. If all frequent itemsets can be computed, then all association rules can be computed easily from the frequent itemsets.

In this paper, we address the question of how to maintain privacy in distributed mining of frequent itemsets. That is, we ask how two (or more) parties can find frequent itemsets in a distributed database without revealing each party's portion of the data to the other. We will formally specify what we mean by "privacy," and our definitions of privacy are cryptographically strong (see Definitions 3 and 4 for details). Roughly speaking, for strong privacy, we require that each participant learns nothing about other participants' data except what is implied by the final output. For weak privacy, we relax the requirement a little to allow that the support count of candidate itemset is leaked to each participant. We will also give solutions for two major types of partitioned data: vertically partitioned and horizontally partitioned (to be defined rigorously in Section 2), respectively, and show that our algorithms preserve privacy.

1.1. Related work

To the best of our knowledge, Clifton and his students were the first to study privacy-preserving distributed mining of association rules and frequent itemsets. In [25], Vaidya and Clifton gave a nice algebraic solution for vertically partitioned data. However, this solution can leak many linear combinations of each party's private data to the other. Furthermore, to process one candidate frequent itemset, its computational overhead is quadratic in the number of transactions. In [18,19], Kantarcioglu and Clifton gave a solution for horizontally partitioned data that uses Yao's *generic* secure-computation protocol as a subprotocol. However, as Goldreich pointed out in [13], generic secure-computation protocols are highly expensive for practical purposes. (In data mining problems, because the input size is huge, they can be even more expensive than in other applications.) Furthermore, the solution in [18,19] only works for three parties or more, not for two parties.[1]

Privacy-preserving data mining has been a topic of active study (see, e.g., papers by Agrawal et al. [2,1]). In particular, many papers have addressed the privacy issues in mining of association rules and frequent itemsets. Some examples are [8,10,23,22,24]. However, these papers are concerned with privacy of individual transactions and/or hiding of sensitive rules. The scenarios they consider are significantly different from ours. We consider a scenario of distributed database, where each part of the data is owned by a different participant. Our target is to protect the private data owned by each participant.

Privacy-preserving distributed mining was first addressed by Lindell and Pinkas [20,21], but their paper only discusses the *classification problem* ("classifying transactions into a discrete set of categories"), not the association rule problem.

As pointed out by Du and Atallah [9], the problems of privacy-preserving data mining can be viewed as an application of generic secure computation. Existing protocols for generic secure computation [27,4,12,6] can solve such problems in theory. However, these generic protocols are highly expensive; and therefore it is our goal to design special-purpose solutions that are much more efficient for certain problems of interest.

[1] In [19], Kantarcioglu and Clifton discussed the difficulty of two parties. In particular, they mentioned that the support count of one participant can be derived by the other participant, by subtracting his own support count from the overall support count. This problem does not exist with our strongly privacy-preserving algorithm in Section 5, because our algorithm does not output the overall support count—it only outputs whether the overall support count is above the threshold.