Applied Soft Computing 13 (2013) 3657–3667

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Incremental learning of privacy-preserving Bayesian networks

Saeed Samet[a,*], Ali Miri[b], Eric Granger[c]

[a] Faculty of Medicine, Memorial University of Newfoundland, St. John's, NL, Canada
[b] Department of Computer Science, Ryerson University, Toronto, Ontario, Canada
[c] Laboratoire d'imagerie, de vision et d'intelligence artificielle, École de technologie supérieure, Université du Québec, Montreal, Canada

ARTICLE INFO

Article history:
Received 26 June 2011
Received in revised form 14 January 2013
Accepted 24 March 2013
Available online 25 April 2013

Keywords:
Security and privacy preserving
Bayesian networks
Incremental learning
Data mining and machine learning

ABSTRACT

Bayesian Networks (BNs) have received significant attention in various academic and industrial applications, such as modeling knowledge in image processing, engineering, medicine and bio-informatics. Preserving the privacy of sensitive data, owned by different parties, is often a critical issue. However, in many practical applications, BNs must train from data that gradually becomes available at different period of times, on which the traditional batch learning algorithms are not suitable or applicable. In this paper, an algorithm based on a new and efficient version of Sufficient Statistics is proposed for incremental learning with BNs. The standard $\mathcal{K}2$ algorithm is also modified to be utilized inside the incremental learning algorithm. Next, some secure building blocks such as secure comparison, and factorial, which are resistant against colluding attacks and could be applied securely over public channels like internet, are presented to be used inside the main protocol. Then a privacy-preserving protocol is proposed for incremental learning of BNs, in which the structure and probabilities are estimated incrementally from homogeneously distributed and gradually available data among two or multi-parties. Finally, security and complexity analysis along with the experimental results are presented to compare with the batch algorithm and to show its performance and applicability in real world applications.

1. Introduction

Bayesian Networks are probabilistic graphical models [24] that are trained to represent the relationship between variables from a dataset [28]. Medical diagnosis applications, fraud detection systems, and financial networks widely utilize such networks to create models and make decision according to the probabilistic independencies among the variables of the underlying databases [29]. For instance, Fenton and Neil in [10] show the successful application of Bayesian Networks in risk management, and Ji et al. in [16] use Bayesian Networks for optimization problems. Also, Yang et al. in [17] utilize Bayesian Networks for optimization in heterogeneous computing environments.

According to the privacy regulations such as Freedom of Information and Protection of Privacy Act (FIPPA) [39] in Canada, or the Health Insurance Portability and Accountability Act (HIPAA) [40] in the United States, individual's private and sensitive data must be secured when protocols are applied on data used to train BNs. To create the BNs structure and parameters using training data which is securely shared among two or more parties, they cannot simply present their own private data to each other, or even to a third party to run a learning algorithm on the whole data. Therefore, privacy-preserving protocols are needed to apply in these situations.

In many practical applications, BNs must be trained using data that becomes available at different points in time. The traditional techniques for training BNs (e.g. $\mathcal{K}2$ algorithm) are batch in nature, and are not suitable for training on data that arrives incrementally. To obtain a high level of performance, using a batch technique would involve accumulating all training data in memory, and recreating a new BN from scratch using all cumulative data. The time and memory complexity of retraining on all data would be prohibitive in applications with large amounts of training data. For instance, selling records in Walmart, as a chain of large stores, are gradually growing everyday and it is not reasonable to store all data and run the data mining and machine learning algorithms on all data every time a block of new data becomes available. This is an ubiquitous scenario in many different fields such as healthcare systems, government applications and so on. Therefore, incremental learning is needed to efficiently update BNs on new data, in terms of data storage and processing time.

In this paper, BNs structure is incrementally constructed each time a block of new training data is available by updating the sufficient statistics of the existing network structure, and a new structure is created accordingly. Note that by updating sufficient statistics, the probability table of each node could also be computed.

* Corresponding author. Tel.: +1 7097778607.
E-mail addresses: ssamet@mun.ca, ssamet42@yahoo.com (S. Samet), Ali.Miri@ryerson.ca (A. Miri), Eric.Granger@etsmtl.ca (E. Granger).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.03.011

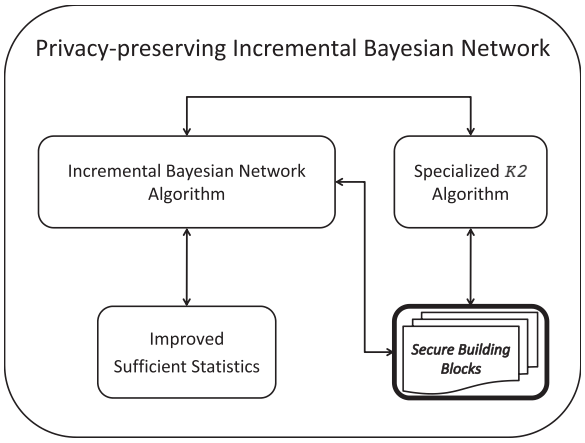

Fig. 1. Overview of the proposed privacy-preserving incremental Bayesian network learning.

After reviewing the existing techniques for incremental learning of BNs, first an improved version of sufficient statistics, in terms of required storage and search time, is proposed followed by a specialized $\mathcal{K}2$ algorithm to be applied in our incremental learning algorithm for BNs. After presenting that algorithm, a privacy-preserving protocol and required secure building blocks, such as secure comparison, and factorial, are proposed along with their security and complexity analysis and experimental results. Fig. 1 illustrates the contribution of the paper and the relations of its components. Inside the main protocol of privacy-preserving incremental BNs, the new incremental learning algorithm is used to update the BNs structure by using an improved version of sufficient statistics and calling the specialized $\mathcal{K}2$ algorithm. Also secure building blocks are utilized inside the protocol to maintain the privacy of the parties involved. These building blocks, which are indicated as a bold box in Fig. 1, are previously proposed by the same authors in [32,33].

As a summary, each time a block of new data becomes available, Incremental BNs algorithm is called with the ordered list of nodes, sufficient statistics of the previous data, the set of previous candidate lists of parents, and the new data. Its output is a directed acyclic graph for the BNs and the new updated sufficient statistics. Inside this algorithm, the specialized $\mathcal{K}2$ algorithm is called when needed with the current node, its parents set, sufficient statistics of that node, its predecessors, and the set of candidate parents lists as the inputs. The output of this algorithm will be the new parents set of the node and its updated candidate parents lists. Each time we need to compute the score function in those two algorithms, secure building blocks are used to securely compute this function without revealing private data of each party to the others.

With this privacy-preserving protocol, it is assumed that data is homogenously shared and owned by several parties, while these parties want to keep their sensitive data private. The protocol is also secure against colluding attacks and could be run over public channels. We show the applicability and efficiency of the proposed protocol by testing it against several different datasets, various number of parties involved and reasonable encryption key sizes, 512, 1024 and 2048 bits, to keep the privacy of the protocol strong. Also, the comparison of the final results of the incremental learning with the batch algorithm, in terms of efficiency and accuracy, shows its great promise to be applied in real world applications.

The rest of the paper is organized as follows. In Section 2, the BN is briefly introduced, along with a brief survey of training algorithms and privacy-preserving BNs. An improved version of sufficient statistics, an incremental algorithm for learning BN structure and a modified $\mathcal{K}2$ [9] algorithm which could exploit that algorithm presented in Section 3. A privacy-preserving protocol for Incremental Bayesian Networks is presented in Section 4, followed by the experimental results in Section 5.

2. Techniques for training Bayesian networks

Bayesian Networks, or Belief Networks, are Directed Acyclic Graphs (DAG) encoding probabilistic relations or dependencies among a set of variables. Each node of a graph represents a variable, and an arc from one node to another node shows a conditional dependency between them. Thus, a BNs structure for a set of variables is formally shown by a pair (N_s, N_p), in which $N_s = (V, E)$ is a DAG containing the set of nodes, V, and the set of edges, E. The set of probability distributions, N_p, which is called the parameters of the BNs, is defined as $N_p = \{p(x_i|\pi_i), x_i \in V\}$, where π_i is the set of x_i's parents and $p(x_i|\pi_i)$ is the probability distribution of x_i conditional upon its parents, π_i.

Constructing BNs is NP-hard [7]. There are different batch algorithms for this learning system. CL algorithm proposed by Chow and Liu [8] estimates the underlying n-dimensional discrete probability distribution from a dataset. To approximate the probability distribution, this algorithm generates the product of $n-1$ second order distributions.

Lam and Bachus [22], and Friedman and Goldszmith [12] use the Minimum Description Length (MDL) principle [21,15] as their approach to propose their own learning algorithm for BNs. In MDL a database is modeled with the minimum length of encoding. Using this approach, BN is encoded as a model with the minimum bit length.

Bouckaert [2] presents a heuristic algorithm, called B, which uses a hill-climbing search method to generate BN structure, and variables do not need to be sorted at the beginning of the algorithm.

Castelo in [5], and Castelo and Cočka in [6] propose algorithm HCMC, which is similar to algorithm B, because of utilizing a hill-climbing search method on DAGs. However, unlike algorithm B it considers the inclusion order among BNs. In [34], two protocols are proposed for privacy-preserving Naive Bayes, in both horizontally and vertically partitioned data using secure multi-party computation techniques such as secure sum [18] and secure dot product [13]. Meng et al. in [25] proposed a privacy-preserving estimation for the BN parameters, by assuming that the Bayesian network structure has been already created, and is publicly known to the parties involved. Also, in [36,37], authors proposed secure techniques to compute the BN parameters on vertically partitioned data using secure multi-party computation sub-protocols, based on their previously presented protocol in [35] which securely developed the BN structure. In [30], authors proposed an approach to learn Bayesian Networks structures from multiple datasets based on the use of Ensembles and an Island Model Genetic Algorithm (IMGA). Another application of Bayesian Networks in security evaluation of networks under attacks has been proposed by Zhang and Song in [38].

$\mathcal{K}2$ algorithm is proposed by Cooper and Herskovits [9]based on a hill-climbing heuristic search to find an optimized BN structure. The $\mathcal{K}2$ algorithm starts with a graph of nodes showing the variables of interest, without any edges. Then, for each node, using a canonical order and a score function, edges by which the score of the graph increases are added as the parents of the current node. This process ends when no more parents can be added or the number of parents reaches a specified threshold, and the next node will be processed in order.

In this paper the $\mathcal{K}2$ algorithm shown in Algorithm 1 is used as a base algorithm for the creation of BNs structure. In this algorithm, π_i is the set of parents of a variable x_i. At each step, score value of the Predecessor nodes of the current node, x_i, is computed and