Engineering Applications of Artificial Intelligence 53 (2016) 1–18

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A sanitization approach for hiding sensitive itemsets based on particle swarm optimization

Jerry Chun-Wei Lin [a,*], Qiankun Liu [a], Philippe Fournier-Viger [b], Tzung-Pei Hong [c], Miroslav Voznak [d], Justin Zhan [e]

[a] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen Graduate School, Shenzhen, China
[b] School of Natural Sciences and Humanities, Harbin Institute of Technology, Shenzhen Graduate School, Shenzhen, China
[c] Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan
[d] Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, Ostrava-Poruba, Czech Republic
[e] Department of Computer Science, University of Nevada, Las Vegas, USA

ARTICLE INFO

Article history:
Received 8 July 2015
Received in revised form
20 March 2016
Accepted 21 March 2016
Available online 6 April 2016

Keywords:
PPDM
Sanitization
Evolutionary computation
Sensitive itemsets
PSO

ABSTRACT

Privacy-preserving data mining (PPDM) has become an important research field in recent years, as approaches for PPDM can discover important information in databases, while ensuring that sensitive information is not revealed. Several algorithms have been proposed to hide sensitive information in databases. They apply addition and deletion operations to perturb an original database and hide the sensitive information. Finding an appropriate set of transactions/itemsets to be perturbed for hiding sensitive information while preserving other important information is a NP-hard problem. In the past, genetic algorithm (GA)-based approaches were developed to hide sensitive itemsets in an original database through transaction deletion. In this paper, a particle swarm optimization (PSO)-based algorithm called PSO2DT is developed to hide sensitive itemsets while minimizing the side effects of the sanitization process. Each particle in the designed PSO2DT algorithm represents a set of transactions to be deleted. Particles are evaluated using a fitness function that is designed to minimize the side effects of sanitization. The proposed algorithm can also determine the maximum number of transactions to be deleted for efficiently hiding sensitive itemsets, unlike the state-of-the-art GA-based approaches. Besides, an important strength of the proposed approach is that few parameters need to be set, and it can still find better solutions to the sanitization problem than GA-based approaches. Furthermore, the pre-large concept is also adopted in the designed algorithm to speed up the evolution process. Substantial experiments on both real-world and synthetic datasets show that the proposed PSO2DT algorithm performs better than the Greedy algorithm and GA-based algorithms in terms of runtime, fail to be hidden (F-T-H), not to be hidden (N-T-H), and database similarity (DS).

1. Introduction

With the rapid growth of information technology and e-commerce applications, it has become increasingly easy to discover useful information and interesting relationships in huge amount of data. Data mining, also called knowledge discovery in database (KDD), provides a set of techniques, commonly used to analyze relationships among purchased products for market basket analysis. Knowledge discovered using KDD techniques can be generally classified as association rules (Agrawal and Srikant, 1994b; Chen et al., 1996; Han et al., 2004), sequential patterns (Agrawal and Srikant, 1995; Mooney and Roddick, 2013; Zaki, 2001), clusters (Murty and Flynn, 1999), and classifications (Quinlan, 1993). Association rule mining (Agrawal and Srikant, 1994b; Chen et al., 1996) is a fundamental KDD task, which consists of discovering interesting information and knowledge in customer transactions.

As data mining techniques can be used to discover implicit information in very large databases, private or secure information may also be easily revealed by those techniques, such as credit card numbers, personal identification numbers, telephone numbers, and other confidential data. Besides, another important issue is that information shared among business collaborators may also be analyzed using data mining techniques to reveal sensitive knowledge that may then be leaked to competitors. Another risk is that a current collaborator may become a competitor, and that this latter may use the strategic knowledge obtained using data mining techniques to take better business decisions, and thus decrease the

* Corresponding author.
E-mail addresses: jerrylin@ieee.org (J.-W. Lin), qiankunliu@ikelab.net (Q. Liu), philfv@hitsz.edu.cn (P. Fournier-Viger), tphong@nuk.edu.tw (T.-P. Hong), miroslav.voznak@vsb.cz (M. Voznak), justin.zhan@unlv.edu (J. Zhan).

http://dx.doi.org/10.1016/j.engappai.2016.03.007
0952-1976/

business performance of the data provider as a result of increased competition. Because of these issues, privacy-preserving data mining (PPDM) has become a critical issue in recent years (Aggarwal et al., 2006; Dasseni et al., 2001; Evfimievski et al., 2002; Lindell and Pinkas, 2000; Verykios et al., 2004). The goal of PPDM is to sanitize a database, that is to hide and secure personal, confidential or sensitive information of participants, while still permitting data analysis. The most common way of hiding sensitive information in a collected database is to sanitize the database through deletion or addition operations. However, this approach may cause several side effects such as hiding non-sensitive patterns, or introducing new artificial patterns. It is a NP-hard (Aggarwal et al., 2006; Verykios et al., 2004) optimization problem to select an appropriate set of sanitization operations for hiding confidential information, while minimizing side effects.

Agrawal and Srikant first introduced PPDM (Aggarwal et al., 2006). Lindell and Pinkas (2000) addressed the issue of PPDM for decision tree learning with the ID3 algorithm. Clifton et al. (2003) presented a toolkit to address various problems in PPDM. To obtain a good balance between data privacy and data utility in PPDM, Pandya et al. (2014) proposed a multiplicative perturbation algorithm. Dwork et al. (2006) generalized previous work by considering both multi-attribute databases and vertically partitioned databases, and designed several algorithms for handling published noisy statistics. Several algorithms were also designed to hide sensitive frequent itemsets or sensitive association rules using custom sanitization procedures (Evfimievski et al., 2002; Hong et al., 2012; Lin et al., 2013; Wu et al., 2007).

Traditional PPDM algorithms have difficulty to cope with the challenge of finding an appropriate set of transactions/itemsets for sanitization that would minimize side effects especially when sensitive information overlaps with important but non sensitive information. Hiding and securing sensitive information may at the same time hide important information. Evolutionary computing is an efficient way of finding near optimal solutions to NP-hard problems. Genetic algorithms (GAs) (Goldberg, 1989; Holland, 1992) are a population-based approach that facilitates the search for good solutions by applying the principles of natural evolution. It has been extensively applied to handle problems having both discrete and continuous variables, nonlinear objectives, and constraint functions without gradient information. In the past, Lin et al. (2014, 2015a) proposed a GA-based algorithm to hide sensitive itemsets using a designed sanitization procedure. In this approach, choosing a set of transactions for deletion is done using a GA framework. It has been shown that GA-based approaches can provide better solutions to PPDM problems with lower side effects compared to traditional Greedy algorithms. Those algorithms still, however, require to manually set the number of transactions to be deleted. Besides, it is a non-trivial task to find appropriate values (rates) for parameters used by GAs such as mutation and crossover rates.

Particle swarm optimization (PSO) was invented by Kennedy and Eberhart (1995). It is inspired by bird flocking to find rich food sources. As GAs, PSO is a population-based search approach, designed to solve optimization problems. In PSO, each particle represents a solution and is evaluated by a predefined fitness function. The personal best (*pbest*) and global best (*gbest*) particles are used to update old particles and generate offsprings of the population, in the evolution process. Since the crossover and mutation operations in GAs are not used in PSO, it is easier to implement the PSO procedure for discovering near optimal solutions. Besides, particles in PSO can transmit information to other particles to speed up the evolution process. In this paper, a PSO-based PSO2DT algorithm is presented to find better sets of transactions to be deleted for hiding sensitive information. The key contributions of the designed algorithm are listed below.

1. In the past, few heuristic approaches have been proposed to sanitize databases for hiding sensitive information. Most of them utilize the GA framework. This is the first paper to address the problem of hiding sensitive itemsets using a PSO-based approach.
2. The designed PSO2DT algorithm is inspired by discrete PSO. It assigns particles and their velocities to set of transaction identifiers, representing transactions to be deleted for hiding sensitive itemsets. An advantage of the designed PSO2DT algorithm is that it has few parameters compared to previous approaches, and it still searches for near optimal solutions to the sanitization problem using a randomized evolutionary approach.
3. The pre-large concept is also adopted in the designed algorithm to avoid performing multiple database scans. This considerably speeds up the evaluation of particles in the evolution process.

The rest of this paper is organized as follows. Related work is reviewed in Section 2. Preliminaries and problem definition are mentioned in Section 3. The PSO2DT sanitization algorithm is presented in Section 4. An example illustrating the proposed algorithm is given in Section 5. Experimental results are reported in Section 6. A conclusion is drawn and future work is discussed in Section 7.

2. Related work

This section reviews related work about GAs, PSO and PPDM.

2.1. Genetic algorithm

In evolutionary computing, population-based approaches are widely used to find near optimal solutions to optimization problems. They are especially used for variations of NP-hard problems and related applications, where it is too expensive to find the best solution by evaluating all solutions. GAs are the most fundamental population-based approach. It has been developed in the early 1970s by Holland (1992). In GAs, a solution is called a chromosome, and it can be evaluated by a designed fitness function. Three operations named selection, crossover, and mutation are used by GAs and are described below.

1. *Crossover*: This operation swaps some bits among two chromosomes (individuals) to generate offsprings of the population. An offspring inherits attributes or characteristics of its two parent chromosomes.
2. *Mutation*: This operation randomly changes one or several bits of an offspring, which may produce variations of its parent characteristics. This operation is used to avoid being trapped in local optimal solutions, and is what allows the evolution process to find near optimal solutions.
3. *Selection*: This operation applies the fitness function to select the best offsprings as the surviving chromosomes. This operation ensures that characteristics of the best offsprings are likely transmitted to the next generation.

The main steps performed by a GA are the following. The first step is to define a type of chromosomes to represent possible solutions. Chromosomes are usually represented as bit strings. An initial population consisting of many chromosomes, also called individuals, is defined, and represents an initial set of possible solutions. The crossover, mutation, and selection operations are then applied to chromosomes to produce the next generation. Each chromosome is evaluated by the designed fitness function to assess the goodness of the chromosomes. This process is then repeated until a termination criterion is satisfied. Although GAs