Chapter 3

A State-of-the-Art Review of Data Stream Anonymization Schemes

Aderonke B. Sakpere
University of Cape Town, South Africa

Anne V. D. M. Kayem
University of Cape Town, South Africa

ABSTRACT

Streaming data emerges from different electronic sources and needs to be processed in real time with minimal delay. Data streams can generate hidden and useful knowledge patterns when mined and analyzed. In spite of these benefits, the issue of privacy needs to be addressed before streaming data is released for mining and analysis purposes. In order to address data privacy concerns, several techniques have emerged. K-anonymity has received considerable attention over other privacy preserving techniques because of its simplicity and efficiency in protecting data. Yet, k-anonymity cannot be directly applied on continuous data (data streams) because of its transient nature. In this chapter, the authors discuss the challenges faced by k-anonymity algorithms in enforcing privacy on data streams and review existing privacy techniques for handling data streams.

INTRODUCTION

The need for data protection, especially when needed for analysis, research and data mining purposes has led to the development of several privacy enforcing schemes. Considerable attention has been given to static data protection (Issa, 2009; Iyengar, 2002; Samarati, 2001; Sweeney, 2001, 2002a, 2002b). Static data are non-real time and so the constraints for processing and/or analysis are not time sensitive. Conversely, there is a lot of data that evolves with time and space, typically referred to as data streams with many real world applications (Guo & Zhang, 2013).

Data streams are real-time and continuous data flows that are ordered implicitly by arrival time or explicitly by timestamps (Golab & Özsu, 2003). The order in which streaming data arrives cannot be pre-determined (Golab & Özsu, 2003). Streaming data emerges from various electronic sources (such as mobile phones or computers) and is expected to be processed online in real-time with minimum...
delay (Zakerzadeh & Osborn, 2013). In Figure 1, we illustrate that streaming data emerge from a source and essentially has a target destination. Examples of applications that use data streaming include web applications, financial applications and security applications (Zakerzadeh & Osborn, 2013). Data streams can also be a form of temporal data (Wang, Xu, Wong, & Fu, 2010). Temporal data is time-critical because the snapshot available at each timestamp must be made available for necessary action (Wang et al., 2010).

Analyzing data streams in real time helps to reveal hidden knowledge and patterns that might need immediate intervention (such intervention could be to detect anomalies in data streams). For example, in many developing nations like South Africa, about 3.3 million crimes occur yearly, this implies that on the average, thousands of crimes occur daily. Suppose the Crime Report Data Stream (CRDS) has the schema: CRDS (VictimName, VictimSex, VictimAge, VictimAddress, CrimeSuffered, SuspectDescription). Analyzing and studying such reported crime in real-time is useful in helping to trap the criminals or suspects in a relatively short period. Furthermore, predictions of future crime or disaster occurrences can be identified. According to Qiu, Li, & Wu (2007), many companies usually outsource the mining of their data to a third party due to lack of in-house expertise. An implication of this is that many law enforcement agencies particularly in developing countries may lack in-house expertise to mine and/or analyze crime reports/data in real time. In order to release the data to a third party, there is a need to ensure proper data anonymization in order to prevent the persons and/or systems analyzing the anonymized data, from identifying the subjects. Accumulating streaming data over long periods can result in delayed predictions and/or reactions thereby subsequently resulting in a late intervention after analysis.

In order to preserve privacy in data streams, a naive approach is to exclude explicit identifiers such as names and/or identification numbers. However, sensitive details about a subject are deducible through linking attacks. Examples of a subject’s sensitive information include crime suffered and medical condition. A linking attack occurs if the combination of non-explicit identifiers (such as date of birth, address and sex) can be used to identify individuals when joined to an external or publicly available table (Li, Ooi, & Wang, 2008; Wang, Li, Ai, & Li, 2007; Zakerzadeh & Osborn, 2013). Non-explicit identifiers used for linking attacks are “Quasi-Identifiers”.

We present a simple example of how a linking attack can occur in Figure 2 by joining publicly available data, Table 1, with a supposedly anonymous streaming data in Figure 2, whose explicit identifier, Name, has been removed. Table 1 is a

Figure 1. Illustration of data streams