Chapter LVI
Privacy-Preserving Data Mining

Alexandre Evfimievski
IBM Almaden Research Center, USA

Tyrone Grandison
IBM Almaden Research Center, USA

INTRODUCTION

Privacy-preserving data mining (PPDM) refers to the area of data mining that seeks to safeguard sensitive information from unsolicited or unsanctioned disclosure. Most traditional data mining techniques analyze and model the data set statistically, in aggregated form, while privacy preservation is primarily concerned with protecting against disclosure of individual data records. This domain separation points to the technical feasibility of PPDM.

Historically, issues related to PPDM were first studied by the national statistical agencies interested in collecting private social and economical data, such as census and tax records, and making it available for analysis by public servants, companies, and researchers. Building accurate socio-economical models is vital for business planning and public policy. Yet, there is no way of knowing in advance what models may be needed, nor is it feasible for the statistical agency to perform all data processing for everyone, playing the role of a trusted third party. Instead, the agency provides the data in a sanitized form that allows statistical processing and protects the privacy of individual records, solving a problem known as privacy-preserving data publishing. For a survey of work in statistical databases, see Adam and Wortmann (1989) and Willenborg and de Waal (2001).

The term privacy-preserving data mining was introduced in the papers Agrawal and Srikant (2000) and Lindell and Pinkas (2000). These papers considered two fundamental problems of PPDM: privacy-preserving data collection and mining a data set partitioned across several private enterprises. Agrawal and Srikant devised a randomization algorithm that allows a large number of users to contribute their private records for efficient centralized data mining while limiting the disclosure of their values; Lindell and Pinkas invented a cryptographic protocol for decision tree construction over a data set horizontally partitioned between two parties. These methods were subsequently refined and extended by many researchers worldwide.

Other areas that influence the development of PPDM include cryptography and secure multiparty computation (Goldreich, 2004; Stinson, 2006), database query auditing for disclosure

detection and prevention (Dinur & Nissim, 2003; Kenthapadi, Mishra, & Nissim, 2005; Kleinberg, Papadimitriou, & Raghavan, 2000), database privacy and policy enforcement (Aggarwal et al., 2004; Agrawal, Kiernan, Srikant, & Xu 2002), database security (Castano, Fugini, Martella, & Samarati, 1995), and of course, specific application domains.

SURVEY OF APPROACHES

The naïve approach to PPDM is "security by obscurity," where algorithms have no proven privacy guarantees. By its nature, privacy preservation is claimed for all data sets and attacks of a certain class, a claim that cannot be proven by examples or informal considerations (Chawla, Dwork, McSherry, Smith, & Wee, 2005). We will avoid further discussion of this approach in this forum. Recently, however, a number of principled approaches have been developed to enable PPDM, some listed below according to their method of defining and enforcing privacy.

Suppression

Privacy can be preserved by simply suppressing all sensitive data before any disclosure or computation occurs. Given a database, we can suppress specific attributes in particular records as dictated by our privacy policy. For a partial suppression, an exact attribute value can be replaced with a less informative value by rounding (e.g., \$23.45 to \$20.00), top coding (e.g., age above 70 is set to 70), generalization (e.g., address to zip code), using intervals (e.g., age 23 to 20-25, name Johnson to J-K), and so forth. Often the privacy guarantee trivially follows from the suppression policy. However, the analysis may be difficult if the choice of alternative suppressions depends on the data being suppressed, or if there is dependency between disclosed and suppressed data. Suppression cannot be used if data mining requires full access to the sensitive values.

Rather than protecting the sensitive values of individual records, we may be interested in suppressing the identity (of a person) linked to a specific record. The process of altering the data set to limit identity linkage is called de-identification. One popular definition for de-identification privacy is *k*-anonymity, formulated in Samarati and Sweeney (1998). A set of personal records is said to be *k*-anonymous if every record is indistinguishable from at least $k - 1$ other records over given quasi-identifier subsets of attributes. A subset of attributes is a quasi-identifier if its value combination may help link some record to other personal information available to an attacker, for example, the combination of age, sex, and address.

To achieve *k*-anonymity, quasi-identifier attributes are completely or partially suppressed. A particular suppression policy is chosen to maximize the utility of the *k*-anonymized data set (Bayardo & Agrawal, 2005; Iyengar, 2002). The attributes that are not among quasi-identifiers, even if sensitive (e.g., diagnosis), are not suppressed and may get linked to an identity (Machanavajjhala, Gehrke, Kifer, & Venkitasubramaniam, 2006). Utility maximization may create an exploitable dependence between the suppressed data and the suppression policy. Finally, *k*-anonymity is difficult to enforce before all data are collected in one trusted place; however, a cryptographic solution is proposed in Zhong, Yang, and Wright (2005) based on Shamir's secret sharing scheme.

Suppression can also be used to protect from the discovery of certain statistical characteristics, such as sensitive association rules, while minimizing the distortion of other data mining results. Many related optimization problems are computationally intractable, but some heuristic algorithms were studied (Atallah, Elmagarmid, Ibrahim, Bertino, & Verykios, 1999; Oliveira & Zaïane, 2003).

Randomization

Suppose there is one central server, for example, of a company, and many customers, each having