Privacy-preserving data mining

A literature review

JOEL BRYNIELSSON, FREDRIK JOHANSSON AND MAGNUS JÄNDEL

FOI, Swedish Defence Research Agency, is a mainly assignment-funded agency under the Ministry of Defence. The core activities are research, method and technology development, as well as studies conducted in the interests of Swedish defence and the safety and security of society. The organisation employs approximately 1000 personnel of whom about 800 are scientists. This makes FOI Sweden's largest research institute. FOI gives its customers access to leading-edge expertise in a large number of fields such as security policy studies, defence and security related analyses, the assessment of various types of threat, systems for control and management of crises, protection against and management of hazardous substances, IT security and the potential offered by new sensors.

FOI
Swedish Defence Research Agency
SE-164 90 Stockholm

Phone: +46 8 555 030 00
Fax: +46 8 555 031 00

www.foi.se

FOI-R-3633-SE
ISSN 1650-1942

Februari 2013

Joel Brynielsson, Fredrik Johansson and Magnus Jändel

Privacy-preserving data mining

A literature review

Titel	Integritetsbevarande informationsutvinning
Title	Privacy-preserving data mining
Rapportnr/Report no	FOI-R-3633-SE
Månad/Month	Februari
Utgivningsår/Year	2013
Antal sidor/Pages	51 p
ISSN	1650-1942
Kund/Customer	Intern
FoT område	-
Projektnr/Project no	I35405
Godkänd av/Approved by	Lars Höstbeck
Ansvarig avdelning	Beslutsstödssystem

Detta verk är skyddat enligt lagen (1960:729) om upphovsrätt till litterära och konstnärliga verk.
All form av kopiering, översättning eller bearbetning utan medgivande är förbjuden

This work is protected under the Act on Copyright in Literary and Artistic Works (SFS 1960:729).
Any form of reproduction, translation or modification without permission is prohibited.

Sammanfattning

Denna studie av informationsutvinning med personlig integritet (PPDM) är baserad på ett kompetensutvecklingsprojekt på 140 timmar. Informationsutvinning extraherar information från data. Det finns ofta en intressekonflikt mellan de fördelar detta ger för företag och myndigheter och personlig integritet. PPDM erbjuder metoder som tar hänsyn både till effektivitet och integritet. I ett inledande avsnitt beskriver vi problemställningen, aktörer och intressen, de olika forskningstraditionerna i PPDM och förhållandet till angränsande forskningsområden. Denna rapport fokuserar på tekniska metoder för PPDM. Det finns två huvudstrategier. Sanerande metoder modifierar data i syfte att både bevara övergripande statistiska egenskaper och ge ett visst mått av integritet. Distribuerade säkra metoder använder kryptering för att beräkna statistiska egenskaper utan att avslöja känsliga detaljer. Det första steget i alla sanerande metoder är att ta bort explicita identifierare som t.ex. personnummer. Detta är vanligtvis inte tillräckligt eftersom individer kan identifieras också genom kvasi-identifierare som förekommer både i måldatabasen och i bakgrundsdata. Sanerande metoder ökar integriteten genom att ta bort kvasi-identifierare. De två huvudsakliga metoderna för detta är 1) deterministisk redigering för att uppnå ett definierat mått av integritet och 2) stokastisk redigering som balanserar statistiska mått på integritet och effektivitet. Olika distribuerade säkra metoder behandlar dels horisontellt uppdelade data där olika parter äger attributuppsättningar för olika personer och vertikalt uppdelade data där attribut som hänför sig till samma personer fördelas mellan olika parter. Översikten kompletteras med några mer udda tekniker och problem, inklusive PPDM för ostrukturerad text och nätverksdata samt metoder för betydelseviktning och klassificerarnedgradering.

Nyckelord: informationsutvinning, personlig integritet

Summary

This review of the research literature in the field of privacy preserving data mining (PPDM) is based on a competence development project spanning over 140 hours of study. Data mining extracts information from data for the benefit of commercial enterprises or governments. There is often a conflict of interest between advantages gained from data mining and privacy. PPDM offers a set of data mining methods that balances the discordant goals of efficiency and privacy. In the introduction we describe the PPDM problem, the main actors and issues, the different research traditions that form the field, and the relation to neighbouring research fields. The focus of this report is technical methods for PPDM. There are two main strategies. Sanitation methods modify data for the purpose of publishing information that both preserves the overall statistical features of the data and offer some degree of privacy. Distributed secure methods use cryptographic techniques to compute statistical measures without revealing privacy-sensitive details. The first step in all sanitation methods is to remove explicit identifiers such as social security numbers. This is typically not sufficient since individuals can be identified also by quasi-identifiers that occur both in the target database and in background data. Sanitation methods increase privacy by removing quasi-identifiers. The two main approaches to this is 1) deterministic editing for the purpose of exactly fulfilling some measure of privacy and 2) randomized editing aiming at balancing statistical measures of privacy and data mining utility. Different sets of distributed secure methods applies to the cases of horizontally partitioned data where different parties own different sets of database records of the same type and vertically partitioned data where data on different attributes pertaining to the same individuals are distributed between different parties. Diverse flavours of distributed secure protocols make different assumptions about the integrity and honesty of the participants. The review of the mainstream methods is supplemented with descriptions of some less often discussed techniques and problems, including PPDM for unstructured text and network data, and the techniques of importance weighting and classifier downgrading.

Keywords: privacy-preserving data mining

Contents

1 Introduction

Privacy-preserving data mining (PPDM) is a nascent research field in computer science focusing on methods for protecting privacy while still enabling data mining. Most of the research in PPDM assumes that one or several databases hold a database *table* consisting of a set of database *records* where each record is related to a specific individual or *record owner* and consists of,

1) Explicit identifiers of the record owner such as name, social security number etc.
2) Sensitive attributes
3) Non-sensitive attributes

In addition there might be *background data*, for example on the Internet, where record owners are associated with a set of background attributes.

The goal of PPDM is to allow data mining without privacy violations where the main types of violations are,

1) Linking the record owner to a sensitive attribute
2) Linking the record owner to a record
3) Linking the record owner to a database table

Linkage means that an adversary finds evidence that the attribute, record or table is associated with an individual with a statistical certainty above a given threshold.

There are several extensions and modifications of this baseline scenario including situations where the database owner is not trusted, data is streamed rather than static, and data describes a social network rather than a set of records.

PPDM methods often assume that the raw data has been filtered by a process of *de-identification* where the explicit identifiers are removed and possibly replaced by synthetic identifiers that cannot be linked to the subjects. Adversaries may, however, have access to background data with further information about the record owners. Combinations

of record attributes can be quasi-identifiers that alone or in combination with background data can be used for inferring the explicit identity of the record owner. If a de-identified data record describes a hospital patient as a 37 years old single farmer from the village of Ölme it is typically easy to identify the record owner from the quasi-identifier by a simple Google search.

The main PPDM methods handle the problem of quasi-identifiers according to two main strategies,

A) Modify the data by suppressing the quasi-identifiers so that privacy is preserved but data mining still is possible. The modified data is published.
B) Keep the intact data in distributed secret databases and use cryptographic techniques for performing data mining without revealing private information.

The process of modifying data for the purpose of privacy is called *sanitation* and produces a *sanitized database*. The data modification strategy is implemented according to two alternative principles,

1) De-identification in combination with purposeful deterministic editing of the initial data so that a given level of privacy is guaranteed while the data mining utility is optimized given the privacy constraint.
2) De-identification in combination with random distortion of the initial data so that a statistical measure of privacy and a statistical measure of data mining utility simultaneously are attained.

The distributed cryptographic approach is also divided in two main cases,

1) Horizontal data distribution where different databases hold different records of the same type.
2) Vertical data distribution where different databases hold different attributes of the same record.

In the following we will discuss PPDM methods from each of the main four categories as well as examples of other maverick methods.

1.1 Actors and issues

Table 1 lists the main players in the context of PPDM together with the issues that is relevant for each actor. Record owners are the entities that are concerned about privacy. They are normally individuals but they could also be groups of people or organizations. Data miners extract useful information from databases and are interested in the utility of the data as well as preserving the trust of the society and being able to prove that legal requirements are fulfilled. Data collectors gather data records and provide possibly sanitized data to database owners. Data collectors often depend on the trust of record owners and must also adhere to legal constraints. The database owner publishes amassed data records to data miners possibly after further sanitation.

Table 1: Actors and issues in PPDM.

Actor	Issues
The international community	Formulate and monitor international agreements
Governments	Formulate and monitor legislation, consumer protection, crime prevention, efficient economy, surveillance and government privileges, ability to trade access to data with other governments
Database owner	Provable ethics and legality, data miner satisfaction, trust of data collectors
Data collector	Provable ethics and legality, access to data and ease of collecting data, value of the collected data
Data miner	Provable ethics and legality, the value of data mining results
Record owner	Privacy, reputation, access to and ownership of personal data, getting notifications of data use and privacy violations

The database owner could have no interest and competence in data mining. For example, hospitals in California publish patient records because such publication is a legal requirement (Carlisle et al., 2007).

Other database owners are very competent and publish data for a specific data mining purpose or perform data mining in-house. The international community and governments are interested in balancing the interests of data miners, database owners, consumers and the general public. Major business and security interests are related to data mining while there is a considerable political pressure for preserving privacy.

1.2 Structure of the research field

The field of PPDM is a confluence of three main research traditions.

The data mining and the statistical disclosure control communities address essentially the same issues but with different terminology and partially overlapping methods. The PPDM problem formulation in this introduction is expressed in the language of the data mining community. The statistical disclosure control community views data records as samples generated by a joint probability distribution and focuses on how filtering data and publishing various marginal and conditional distributions impact on privacy where privacy is measured by the probability of revealing private information.

The distributed cryptographic approach originates in the field of secure distributed computation where the goal is to enable mistrustful participants to jointly perform computations. The methods in this field are essentially applications of cryptography.

1.3 Related research fields

Privacy-preserving data publishing (PPDP) is concerned with methods for publishing data so that privacy is preserved independently of the data mining methods that may be applied on the published data (Fung, 2011). PPDP is concerned with hiding the identity of the record owners rather than hiding the sensitive data per se.

Statistical Disclosure Control (SDC) focus on privacy-preserving publishing of statistical tables including means for avoiding direct revelation of subject identity and record attributes as well as avoiding inference of identities and attributes with statistical confidence above a set level.

Privacy-preserving data collection (PPDC) assumes that subjects do not trust the database owner and therefore employs cryptographic methods to collect data records without revealing the record owner´s identity (Yang et al., 2005; Jakobsson et al., 2002).

2 Technical background

2.1 Notation

Vectors are written in bold type e.g. **x.** The i:th component of the vector **x** is written x_i.

2.2 Association rule mining

The purpose of association rule mining (ARM) is to find statistically relevant relations between attributes in a database. Consider an online store selling items from the inventory $\{I_1, I_2, ..., I_n\}$. The transaction database consists of records where each record indicates that an identified customer has purchased a set of items. An association rule has the form $X \Rightarrow Y$ where *X* and *Y* are sets of items selected from the inventory. The meaning of the association rule is that customers purchasing items *X* are inclined to also purchase items *Y* in the same transaction. The rule is often quantified by two measures: support and confidence. The *support* of a set of items *X* is the proportion of transactions where *X* is a subset of the items in the transaction record. Hence, the support measures how common it is to find the prerequisites for applying the rule satisfied. The *confidence* in an association rule $X \Rightarrow Y$ is the fraction of transactions where *Y* is a subset of the items in the transaction record given that *X* also is a subset of the items in the transaction record. The confidence estimates hence the probability that customers purchase *Y* provided that it is known that they purchase *X*. Ideally, data miners want to find rules with high support and high confidence although rules with some support and low but significant confidence also can be very useful. An online store may use an association rule $X \Rightarrow Y$ to offer products from the itemset *Y* to customers that already have selected products in the itemset *X*. This can drive significant sales even if the confidence in the rule is low.

3 Privacy-preserving methods

3.1 Randomization

The main references for this section are Chapters 6 and 7 in Aggarwal and Yu (2008).

3.1.1 Tutorial example

Consider a stream of data records being generated by user activities on a web site maintained by some company offering products or services through the web. Each record contains the activities that are generated by one identifiable web site user, i.e., a record holds a list of sequential time-stamped events or “clicks” that a web site user has performed. The typical user of interest can be thought of as a potential customer browsing a company’s web site to gain information and buy services or products. After buying something, the user might be interested in buying related products or to find out more about the product or perhaps the company’s offerings at large due to some advertisement campaign. Hence, to optimize their business and/or the customer experience, the web company would benefit from analysing the users' event history to learn about user behaviour. Examples of such analysis would be to use association rule mining for finding browsing behaviour preceding a likely purchase in order to offer the relevant product earlier in time, or to use classification techniques in order to assign a label to the customer.

A second business-to-business (B2B) company has specialized in helping online companies to analyse the event history of web site users. This company specializes in performing data mining and providing descriptive statistics based on users' event history that is obtained in real-time. Hence, the customers of this second B2B company consist of a number of online companies who continuously wish to improve their business based on the actions taking place on their web pages. This situation is frequently occurring in today's e-commerce business where, technically, the information is passed directly from the online store's web page to the analysis company through the use of web pages containing “invisible images” (typically

images consisting of a single pixel, having the same colour as the web page background colour) that are physically located on the analysis company's web server, i.e., the web page at the online store contains objects residing on the analysis company's web server.

Of course, for any company to give away intelligence regarding their customers' behaviour is unacceptable, so the data stream coming from the web company to the analysis company must be perturbed in a way that preserves the user privacy whilst still making it possible for the analysis company to come up with useful results. Randomization is a perturbation method which is particularly suited for these kinds of situations since the added noise is independent of the other records and, henceforth, can be added in an iterative manner.

3.1.2 Background

Randomization is a form of privacy-preserving data publishing, i.e., a way to perturb data records to preserve privacy whilst retaining the data usefulness for data mining purposes. For the most part, the individual records can never be recovered. Instead, the idea is that representative distributions of the records can be recovered and used for data mining purposes.

3.1.3 Method overview

Randomization comes in two main flavours when it comes to distortion techniques: additive and multiplicative addition of noise. In the additive randomization version, a random noise value is drawn independently from a probability distribution and added to the data record attribute. Multiplicative addition of noise is instead performed by a projection of the attribute vectors using a suitable matrix multiplication.

3.1.3.1 Additive perturbation

Let X, Y, and Z be random variables representing the original data records, the noise to be added, and the resulting distorted records, respectively. Now, by considering samples or the distributions of these three variables, it is possible to describe the process of adding random noise to preserve privacy, how to use the distorted records for data mining purposes, and how to analyse possible attacks.

The process of adding random noise to a data record can be described as follows. Let $x_i \in X$ be a data record to be perturbed. Draw a noise value $y_i \in Y$ from the known distribution f_Y and add this noise to x_i in order to obtain the distorted record $z_i = x_i + y_i$. Now, after perturbing a set or stream of N original data records $x_1, \ldots, x_N$ we also know N samples $z_1, \ldots, z_N$ of Z and can approximate the distribution of the perturbed records. Also, since $Z = X + Y$ we have that $X = Z - Y$ which makes it possible to obtain N samples of X by subtracting away new noise values drawn from f_Y. Hence, given that the variance of the added noise is large enough, the general idea is that the original records $x_1, \ldots, x_N$ cannot be recovered, but the distribution containing the behaviour of X can be recovered. Note, however, that the process of first approximating Z and then subtracting away Y is not desired since errors in the estimation of Z then may get enlarged after subtracting away Y (i.e., the errors add up each time a random variable is approximated). Instead, the randomization method needs to be considered along with an iterative reconstruction method where the process of approximating Z and subtracting away Y is combined. The choice and development of such reconstruction methods in order to approximate the distribution of X optimally is at the heart of the additive randomization method. Two examples of such iterative reconstruction methods are the Bayes and the EM reconstruction methods where the latter has been shown to perform optimally with regard to several desired properties (Agrawal & Aggarwal, 2001).

Since the addition of noise is performed independently for each attribute, the distribution reconstruction results in a number of uni-variate distributions describing the behaviour of the record attributes (rather than one single multi-variate distribution describing the data records). The reason for this is mainly due to that the identity of a record in a reconstructed multi-variate distribution could be guessed by correlating the distances between the attribute values. Hence, new data mining algorithms taking multiple uni-variate distributions into account need to be developed, since “traditional” data mining algorithms cannot be used.

Two methods for reconstructing the original data record distributions, i.e., to approximate f_X and F_X, are the Bayes reconstruction method and the EM reconstruction method. Bayes reconstruction, which is actually an approximation of the EM reconstruction method, uses Bayesian updating where f_X is initially set to a uniform distribution which is iteratively refined within a number of intervals (i.e., when new data arrives, f_X is refined at a number of fixed points and therefore needs to be discretized).

As mentioned, the aggregate behaviour (sum, count, average, maximum, minimum, percentile values, etc.) of the attribute distributions is what can be reconstructed and used by data mining algorithms using the randomized data. For example, Agrawal and Srikant (2000) discuss a classification method using decision trees where the splitting points are defined separately by the attribute distributions.

Attacking additive randomization can be more or less easy depending on the underlying original data records. In some circumstances, such as when the density function f_X is well approximated (i.e., given many data points) and concentrated to intervals separated by zero density intervals, one can immediately compromise the privacy by identifying an attribute value to a certain density interval. Another possible attack is to use public information for determining the identity of a perturbed record by trying to fit the "potential perturbation" to the perturbation distribution f_Y, i.e., for a public attribute w and a perturbed attribute z we investigate whether $z-w$ fits f_Y. In cases where there are many dimensions/attributes and it is known that the public dataset indeed includes the record to search for, a maximum likelihood fit can identify the record with a high degree of certainty.

Overall, the additive randomization technique for performing PPDM brings about a natural trade-off between information loss and privacy, which must be taken into account. The privacy goal is that individual record values cannot be recovered, but this has to be balanced against the utility goal that the information provided by the distribution should be useful. In general, the added noise needs to be sufficiently large but still make it possible to recover and use the aggregate data distribution.

3.1.3.2 Multiplicative perturbation

In its additive version, the randomization approach focuses on single-dimensional perturbation and assumes independent database columns. By using multiplicative perturbation, it is possible to preserve some of the inter-attribute properties by multiplying the attributes with a random noise matrix which, hence, provides a random projection of the original attribute vector. Put simple, multiplicative perturbation is about using linear algebra matrix transformations on the whole, or at least whole parts, of the dataset in order to preserve distance relationships inherent in the data. That is, the attribute dimensions are kept dependent so that data mining possibilities persist. Hence, multiplicative perturbation can be seen as a way to improve the privacy whilst preserving the mining task and the model-specific data, which differs from additive perturbation. In other words, instead of finding a balance between the level of privacy and the level of utility as in additive randomization, multiplicative perturbation serves to maintain the level of utility while improving the level of privacy.

Compared to non-multiplicative multi-dimensional data perturbation, the idea of using matrix transformations for performing multi-dimensional data perturbation makes it possible to preserve inter-data relationships regarding, e.g., distances between vectors, inner products, etc. Hence, depending on the data mining need, one preserves the measures needed to perform the mining task. For example, a k-nearest neighbour (k-NN) classifier performs classification based on the labelling of the k nearest neighbours, which would be the same neighbours also after perturbation, given that the distances between the data points are kept.

The simplest form of multiplicative perturbation is *rotation perturbation* which covers not only rotations but all kinds of transformations that can be achieved using an orthonormal transformation matrix $R_{d\times d}$ where d is the number of dimensions/attributes. From linear algebra, we recall that an orthonormal/orthogonal matrix is a square matrix where both rows and columns consist of orthogonal unit vectors (i.e., orthonormal vectors). These matrices can be used for linear transformations of vectors that preserve the dot product between the vectors, i.e., transformations such as rotations, reflections and other kinds of axis permutations. Hence,

rotation perturbations make it possible to preserve the Euclidean distance between records along with other geometric properties in the dataset. Given a data record matrix $X_{d\times N}$ with N records and d dimensions/attributes, rotation perturbation can either be applied to the whole dataset or to parts of the columns (i.e., depending on the mining task, we might only need to maintain certain attribute correlations).

Using *projection perturbation*, the idea is to reduce the number of dimensions by using a projection matrix $P_{k\times d}$ where $k < d$. Again recalling some linear algebra, we note that multiplying a $k\times d$ matrix with a $d\times N$ matrix produces a $k\times N$ matrix so the idea here is to reduce the number of attributes by linearly projecting the original attribute values on a fewer number of attributes to still approximately preserve the distance relationships between the columns. It has been proven (Johnson & Lindenstrauss, 1984) that it is possible to find approximations that preserve the distances well, but although methods have been suggested (Liu et al., 2006) it remains challenging to generate these desired projection matrices in practice.

Finally, *geometric perturbation* is a direct extension of the orthonormal transformation that is used in rotation perturbation, aiming to come up with a more attack-resilient perturbation approach that still exhibits some of the advantageous properties of the rotation perturbation approach. Here, two additional noise components are added in order to 1) perturb the transformation, 2) distort the individual attribute values independently of each other. To perturb the transformation, a random vector $\mathbf{t}_{d\times 1}$ is used to create a ``translation matrix" $\Psi = [\mathbf{t},\mathbf{t},\ldots,\mathbf{t}]_{d\times N}$. To distort the individual values, a random noise matrix $\Delta_{d\times N}$ is used, where each element in $\Delta_{d\times N}$ is independently and identically distributed. Given an orthonormal transformation matrix $R_{d\times d}$ and a data record matrix $X_{d\times N}$ with N records and d dimensions/attributes (as in rotation perturbation), geometric perturbation can now be defined as $RX + \Psi + \Delta$. The idea behind adding the translation matrix noise Ψ is that the distance between attribute vectors is not changed since $\|(\mathbf{x}+\mathbf{t})-(\mathbf{y}+\mathbf{t})\| = \|\mathbf{x}-\mathbf{y}\|$ while the rotation origin is indeed changed so that it is more difficult to attack by exploiting records close to the

rotation origin. It should be noted, though, that the translation matrix operation indeed alters the inner product between the vectors. Regarding the random noise matrix Δ the idea is that a low intensity noise protects from attacks trying to infer the distance information whilst still maintaining the data record usefulness with regard to, e.g., class boundaries and cluster membership.

In contrast to additive randomization, the key idea when using multiplicative perturbation is to look for (existing) transformation-invariant models and algorithms instead of having to design new data mining algorithms for the tasks at hand. As an example, given a transformation function $T(X)$ that transforms a dataset X into X_T, we think of a classifier f to be invariant to the transformation $T(X)$ given that $f_X(Y) \equiv f_{T(X)}(T(Y))$ holds for all training datasets X and test datasets Y. As already mentioned, a kNN classifier is an example of a classifier being entirely dependent upon distance, and can therefore be considered to be invariant to both rotations and translations (this example extends to similar kernel methods in general).

Attacks on multiplicative perturbation include exploiting the rotation centre, i.e., using the fact that points close to the origin will still be close to the origin after the rotation perturbation. A random translation perturbation, as performed in a geometric perturbation, addresses this problem by hiding the rotation centre. More effective attacks targeting both the rotation and the translation aspects could be launched using the independent component analysis (ICA) technique that iteratively reconstructs the attribute vectors in the original dataset. For some specific datasets exhibiting dependencies between the row vectors in the source matrix, an ICA attack can be very efficient and also breaks the basic rotation perturbation totally. However, for the generic case the method is difficult to apply without having additional knowledge about the distribution of the original dataset. Even more sophisticated attacks that exploit the distance relationships in the original dataset in order to infer the rotation and translation matrices also exist, but would require considerable additional prior knowledge about the source data records. Protecting from these kinds of distance-based attacks include the addition of random noise according to the geometric perturbation method.

At the end of the day, evaluating and subsequently choosing a particular perturbation that is as attack-resilient as possible becomes important in order for the multiplicative perturbation technique to be useful. Considering evaluation, the variance-of-difference (VoD) approach measures the difference between the original attribute column and an estimated column in terms of a random variable where the variance denotes the difficulty in estimating the original attribute data. Now, the VoD measure can be used in combination with the described attack methods to develop a hill-climbing method that samples random translation matrices that are further refined using local maximization of the VoD measure (Chen et al., 2007).

3.1.4 Comments and conclusions

A key property of the randomization method is the generic properties of the method, i.e., that noise is added to the data records without considering the data content. This is useful since noise can be added already at data collection time without having to store or wait for the whole database table. However, for the same reason, the method does not account for the re-identification risk caused by outlier records (that are difficult to mask by this method) or information gained from public data. In contrast, k-anonymity (described further in Section 3.2) holds the opposite properties by using the content of the data records to provide a certain amount of privacy guarantee.

An assumption being made is that one is not allowed to perform any kind of learning or remembering of precise data records. This is indeed a challenge for additive randomization since only the distributions are known for performing data mining on the underlying data. This limits the range of algorithmic techniques that one is able to use (since most standard data mining algorithms cannot deal with such input). Another problem with additive randomization is that of bias, meaning that a query using the perturbed data results in a significantly different result than the query would have done if it would have used the original data. Several types of biases exist due to, e.g., variance changes in individual attributes, changes in relationships between different (confidential or non-confidential) attributes, distribution changes, etc. (Muralidhar et al., 1999).

Attacks on additive randomization include that of exploitation of the underlying distribution of attribute data and that of using public information. Regarding multiplicative perturbations, they are specifically designed for the purpose of preserving relationships so this is also the key to attacking the method. Here, attacks can use knowledge about the original record vectors that can be exploited using linear algebra or principal component analysis.

3.2 *k*-anonymity

The main reference for this section is chapter 5 of Aggarwal & Yu (2008).

3.2.1 Tutorial example

Consider a large medical database, consisting of patient information including identifiers such as social security number and full name, non-sensitive attributes such as ZIP-code and gender, as well as more sensitive attributes such as the obtained diagnoses of the patients. The database owners would like to make parts of the dataset public in order to allow other researchers to make new discoveries (such as finding likely contributing factors for some of the diseases). However, due to ethical and legal reasons, they would first like to make sure that the published records cannot be traced back to specific individuals. This is a situation where *k*-anonymization comes in handy. The database owner chooses a suitable value of *k* (in this example case *k*=5), removes all identifiers that uniquely can identify an individual, and applies a *k*-anonymization algorithm. The algorithm reduces the granularity of the data (by generalizing and/or suppressing data) in such a way that it is not possible to distinguish a given record from at least *k*-1=4 other records in the resulting public database. The resulting database is thereafter published, while the original database is kept private.

3.2.1.1 Background

While it is clear that unique identifiers such as social security number or passport number have to be removed if it should not be possible to directly find out to which individual a record (including sensitive information) refers, it may not be equally obvious that also non-

sensitive and non-unique information such as gender, age, or ZIP-code can be used to uniquely identify an individual. However, by combining a few such attributes (referred to as quasi-identifiers or key attributes), it is in many cases possible to perform re-identification. As an example, it may not be possible to uniquely identify which individual a certain record containing the attributes date-of-birth, gender, and place-of-birth belongs to for a record in which the value of the place-of-birth attribute corresponds to a large city, but it is not improbable that the same information is enough to uniquely re-identify a certain individual if the place-of-birth refers to a small village. As will be described in the following, the purpose of *k*-anonymity is to prevent such re-identification.

3.2.2 Method overview

K-anonymity can be seen as a method for privacy-preserving data publishing. The objective of the method is as already has been mentioned to prevent re-identification of individuals by ensuring that a record cannot belong to less than *k* distinct individuals. More precisely, *k*-anonymity can be defined in the following way:

"Each release of the data must be such that every combination of values of quasi-identifiers can be indistinguishably matched to at least k respondents"

In order to assure this, there are two kinds of methods available for reducing the granularity of the data: generalization and suppression. These methods preserve the truthfulness of the data, which is an advantage in comparison to e.g., randomization. Examples of generalization are to replace the ZIP-code 531 41 with 531* or to replace the marital status *divorced* with *been married* (including both married and divorced people), while an example of suppression is to remove a record or an attribute from the database. Suppression can e.g., be useful for removing a few outliers that otherwise would require a high degree of generalization (and thereby reducing the information loss). A general approach to ensure *k*-anonymity for any value of *k* in any record in any database would be to suppress all data, but this is obviously not a very useful approach since the utility would be 0 even though the obtained privacy would be maximal. Furthermore, for a given value of *k*, there will be many feasible solutions for obtaining the

wanted privacy level. These solutions can give various utility, and optimal k-anonymity would correspond to finding the feasible solution that maximizes the utility (sometimes referred to as *k*-minimal generalization). In Meyerson et al. (2004), it has been shown that optimal *k*-anonymization is a NP-hard problem, making it infeasible to solve optimally for reasonably large databases.

One example of an optimal algorithm for *k*-anonymization is the K-Optimize algorithm suggested by Bayardo & Agrawal (2005). This is a branch-and-bound algorithm which also can be used as a non-optimal algorithm (by setting a maximal computational time limit and returning the current best solution if this limit is reached). There are also many purely heuristic algorithms suggested in literature that performs well (even though they cannot guarantee optimal solutions). In fact, *k*-anonymity can be seen as a search over a space of multi-dimensional solutions, making it possible to use any standard heuristic algorithm such as simulated annealing and genetic algorithms.

In the above, we have assumed that we have a situation where the database owners would like to publish data on which other people can apply various statistical tests or data mining algorithms to infer new knowledge. If it is anyone else than the database owner that will apply data mining algorithms and we would like to guarantee *k*-anonymity of the obtained results it is necessary to apply the *k*-anonymization process first and do the data mining afterwards. However, there may also be a situation where the database owners will do the data mining and would like to ensure that the obtained results guarantee *k*-anonymity. In this case, we can still apply the *k*-anonymization first and then apply the data mining algorithm in a second step, but this may give less useful results than necessary. As an alternative, we can do the data mining first and in a second step make sure that the results fulfil *k*-anonymity. Another option is to apply special data mining algorithms that perform both steps in the same process. The two last types of mine-and-anonymize methods are in fact often preferable to anonymize-and-mine methods, given that the database owners can make the data mining themselves.

3.2.3 Comments and conclusions

One potential type of attack against databases that have been "anonymized" by removing unique identifiers is to use combinations of quasi-identifiers still in the data and to match the values of those with the values for the corresponding attributes in public records or other kinds of background information. The presented *k*-anonymity model protects against such attacks by assuring that no individual can be identified with a certainty exceeding 1/*k*. However, this does not mean that it is guaranteed that no sensitive information about certain individuals can be obtained from a *k*-anonymized database. So called homogeneity attacks can be made, in which a lack of diversity among sensitive attribute values is exploited by the attacker (one example would be an equivalence class of *k* records where all share the same disease). To counter such problems, the *l*-diversity model has been suggested as an extension to *k*-anonymity. Such a model requires that in each equivalence class (consisting of *k* or more records) there are at least *l* "well-represented" sensitive values so that there is an intra-group diversity protecting against homogeneity attacks.

A problem with both *k*-anonymity and *l*-diversity is that the methods are sensitive to the curse of dimensionality. For high dimensional datasets, suppression of a large number of attributes is often needed for guaranteeing *k*-anonymity. Obviously, such suppression heavily reduces the utility of the data. The increased dimensionality also makes the problem more difficult from a computational perspective. The situation is even worse for *l*-diversity. If there are several sensitive attributes, *l*-diversity in many cases becomes very hard to achieve at all.

As a final note on *k*-anonymity and *l*-diversity it can be mentioned that both methods give all records the same amount of protection, i.e., it is not possible to decide that different tuples should have different degrees of privacy preservation in its standard version. Approaches to augment *k*-anonymity and *l*-diversity with personalization features have however been suggested (the interested reader is referred to chapter 19 of (Aggarwal and Yu, 2008)), making it possible for individuals to personalize what value of *k* they demand, or to formulate semantically-richer privacy preferences.

3.3 PPDM for distributed databases with horizontal data partitioning (PPDDM-H)

The main reference for this section is chapter 13 of Aggarwal and Yu (2008).

3.3.1 Tutorial example

Consider an international operation where the three participants wish to compile the total monthly petrol consumption for the purpose of dimensioning transport facilities. None of the participants is, however, willing to reveal their own petrol consumption since this provides intelligence on their operations. There is no mutually trusted party. A PPDM method can solve this problem as follows. Participant A generates a private and a public encryption key and sends the public key to participants B and C. Participant B uses the key to encrypt the value of its own petrol consumption and sends the result to participant C. Participant C encrypts the value of its own petrol consumption and applies a special method, to be discussed in the following, to both encrypted values for the purpose of computing the encrypted value of the sum of B's and C's fuel consumption. This encrypted sum is sent to participant A. Using the private key, participant A decrypts the received encrypted value and recovers the sum of B's and C's fuel consumption in clear. Participant A adds its own fuel consumption and broadcasts the value of the total fuel consumption of the coalition. Since participants B and C are unable to decrypt, everyone learns the total value and nothing else.

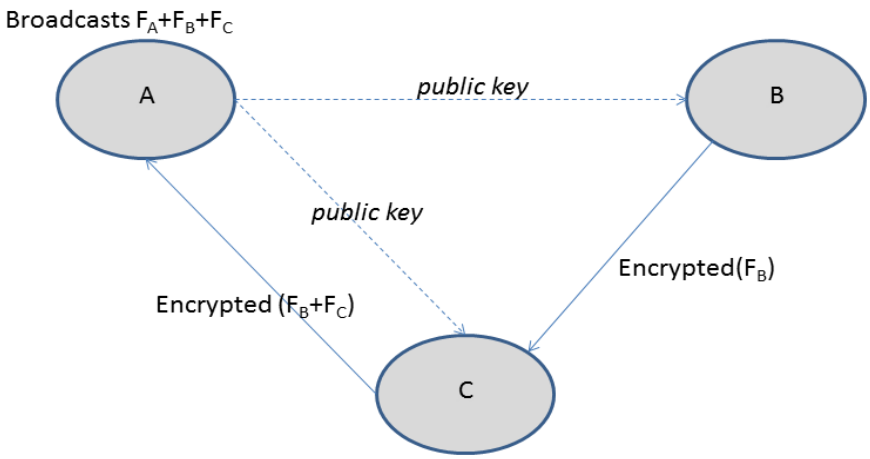

Fig. 1. A simple example of PPDM for distributed databases. F_A, F_B and F_C denote the monthly fuel consumption of participants A, B and C respectively.

3.3.2 Background

PPDDM-H algorithms consist of three hierarchical layers,

1) Homomorphic encryption techniques

2) Secure sub-protocols

3) Application algorithms

Each layer has its own research literature. This section describes layers 1 and 2. It is assumed that the reader knows the basics about public key cryptosystems (Katz & Lindell, 2007).

3.3.2.1 Homomorphic encryption

The methods that are discussed in this section use a public key encryption technique termed *Homomorphic encryption*, which allows operations such as summation and multiplication on encrypted data. The input to the homomorphic addition operator is $E_{pk}(v_1)$ and $E_{pk}(v_2)$ - the homomorphic encryptions with the key *pk* of plaintext values v_1 and v_2 respectively. The output of the addition operator is $E_{pk}(v_1 + v_2)$, the encryption of $v_1 + v_2$ using the same key *pk* as for the inputs. The input to the homomorphic multiplication operator is k and $E_{pk}(v_1)$ where the former is a numerical constant. The output of the multiplication operator is $E_{pk}(k \cdot v_1)$ the encryption of $k \cdot v_1$ using the same public key *pk* as for the inputs. By combining these basic operations it is possible to build algorithms on encrypted data without knowledge of the decryption key.

3.3.2.2 Secure sub-protocols

Secure sub-protocols use the basic operations of homomorphic encryption and are used as building blocks for PPDDM-H application algorithms. Important secure sub-protocols are secure summation, secure comparison, secure dot product, secure polynomial evaluation, secure logarithm, secure intersection and secure set union.

As an example we will provide a more detailed description of the secure sub-protocol for secure summation. A simplified version of this protocol was sketched in the tutorial example. Consider m sites each holding a cleartext value v_i. The objective of the secure sub-protocol is to compute $V = \sum_{i=1}^{m} v_i$ without revealing anything else than V to any of the participants. The protocol works as follows.

1) Site 1 creates homomorphic keys, distributes the public encryption key *pk* to all other sites and keeps the private decryption key secret.
2) Site 2 encrypts its cleartext value and sends the result $E_{pk}(v_1)$ to site 3.

3) All sites $3 \leq s \leq m$ gets $E_{pk}(\sum_{i=2}^{s-1} v_i)$ from the previous site *s-1* and applies the homomorphic addition operator to compute $E_{pk}(\sum_{i=2}^{s} v_i)$ which is transmitted to the next site *s+1* or in the case of site *m* back to site 1.

4) Site 1 receives $E_{pk}(\sum_{i=2}^{m} v_i)$ and decrypts to cleartext $\sum_{i=2}^{m} v_i$. After adding v_1, the intended result *V* is distributed – usually to all participants.

All participants learn the value of *V* but no other information is gained from participating in the protocol. Site 1 has not a privileged position. Consider what happens if one of the participant is malicious. By violating the protocol site 1 could distribute a misleading value of *V* and keep the real value of *V* for private use. Participants 2, 3, …, *m* can also cause a misleading value of *V* by inserting an incorrect value and would also be able to compute the real value of *V*.

3.3.3 Method overview

3.3.3.1 Objectives

Consider a situation where multiple data sources hold different records of data of a common type i.e. *horizontal data partitioning*. The owners of the data sources are in general reluctant to share data but are willing cooperate for the purpose of generating global statistics of common interest. Companies in the same line of business could for example each hold a database of sales information where each record includes the same type of information. In spite of the obvious need for privacy, companies might be interested in cooperating in compiling overall industry statistics. It is often not possible to find a trusted intermediary that is allowed to process the combined raw data.

The objective of *PPDM for distributed databases with horizontal data partitioning* (PPDDM-H) is to enable global data mining in distributed databases without trusted intermediaries and without disclosing any

information to any of the participants beyond the intended result of the mining operation. Note that the intended result often reveals information beyond the face value of the result. Combining the intended result with background information can give further information. All participants in the tutorial example can for example compute the sum of the other parties' petrol consumption. Adding some background information might enable a good estimate of the other participant´s fuel use. The PPDDM-H methods are, however, only concerned with what the participants can learn directly from performing the PPDDM-H protocols.

PPDDM-H methods are specified to handle a given *adversarial model* that defines the degree of trust between the participants. Basic adversarial models are,

- The *Semi-Honest* model where participants can be trusted to follow the agreed protocol precisely but may try to use information that is made available by the protocol for learning about the other participant's confidential data.
- The *Malicious* model where participants could violate the agreed protocol individually or in complicity with other malicious parties.

3.3.3.2 Applications

The baseline secure protocols can be combined into many different types of PPDDM-H algorithms including classifiers (decision tree, naïve Bayes, nearest neighbour, support vector machine) and clustering algorithms (k-means, expectation maximation). In the following we will explain PPDDM-H in the context of association rule mining. A brief introduction to association rule mining is found in section 2.2. A simple method for PPDDM-H association rule mining is,

1) Each site computes local candidate rules with support and confidence that exceeds agreed thresholds. The sub-protocol *secure set union* is used to merge the local sets of candidates without revealing anything other than the combination of all local candidate sets. Any rule with global support and confidence above thresholds

is guaranteed to be in the merged candidate set since any such rule must fall above thresholds in at least one local context. The rules in the resulting global candidate set can now be evaluated.

2) The global support for each candidate rule $X \Rightarrow Y$ can be computed using the secure addition sub-protocol. The total number of transactions of any type is compiled by securely adding the local number of transactions of any type. The total number of transactions including *X* is calculated by securely adding the local number of transactions including *X*. The global support for the rule is the ratio between the total number of transactions including *X* and the total number of transactions of any type.
3) The global confidence of each candidate rule $X \Rightarrow Y$ is obtained by secure summation of the total number of transactions including $X \cup Y$ followed by division by the already globally available total number of transactions including *X* .
4) Globally valid rules are selected from candidates with support and confidence above thresholds.

The algorithm described here serves to illustrate the concept of PPDDM-H but is a gross simplification of the more sophisticated secure association rule mining methods that can be found in the literature where improvements include both efficiency and privacy.

3.3.4 Comments and conclusions

Repeated application of secure sub-protocols could expose private data in spite of the formal security of each individual application. Consider for example the secure dot product protocol that is used extensively in the learning process of support vector machine classification. To illustrate a simple *probing* attack method, we assume that there are two sites each holding local values **x** and **y** of a vector with *m* components. The secure dot product protocol enables the global computation of, $\mathbf{x} \bullet \mathbf{y} = \sum_{i=1}^{m} x_i y_i$ without revealing anything else than the final result.

Within the legal protocol of a support vector machine application, site 1 could now maliciously request repeated evaluations of the dot

product $\mathbf{x} \bullet \mathbf{y}$ while each time supplying a judiciously chosen value of **x**. If the sequence of **x** values form base vectors of the vector space spanned by the possible values of **y** , site 1 would be able to deduce the value of **y** from the dot products. Since a support vector machine learning process (using linear kernels) could include repeated dot products between a given training vector at site 2 and many different training vectors at site 1, it would be difficult for site 2 to discover the malicious behaviour.

Present PPDDM-H protocols do not support trade-off between privacy and utility although there is some research (Feigenbaum et al., 2006) on methods that delivers an approximate global result and thus offer increased privacy.

Because of the bucket brigade communication model, PPDDM-H often scales linearly as the number of sites increases. The main efficiency issue in PPDDM-H is the computational cost of cryptographic protocols. In current PPDDM-H protocols there are two approaches to achieve trade-off between privacy and efficiency. Firstly, the assumed adversarial model should reflect a proper balance between privacy concerns and efficiency. Secondly, the choice of cryptographic algorithm and lengths of cryptographic keys can be adapted.

PPDDM-H methods can only be applied in structured collaboration between organizations (companies, armed forces, governments). Data collectors and record owners must trust the organizations that handle the data. If PPDDM-H methods are improved and standardized in the future, it would be possible for governments and international authorities to enforce their use.

3.4 PPDM for distributed databases with vertical data partitioning (PPDDM-V)

The main reference for this section is chapter 14 of Aggarwal and Yu (2008).

3.4.1 Tutorial example

To illustrate the concept of PPDDM-V consider two organizations that each have data about a group of 1.000.000 people. The police know if

the subjects have a criminal record and a health insurance company knows if the subjects have a history of depression. The subjects are identified by social security number and each list is in numerical order. A scientist wants to compute the probability of having a criminal record given that a subject has a history of depression. Each organization refuses external access to its own data set. Using a PPDDM-V technique it is, however, still possible to compute both the number of depressed people with a criminal record and the total number of depressed people so that the scientist can estimate the requested probability.

The police encrypt their data set, sends the result to the health insurance company for a second layer of encryption after which the doubly encrypted data is forwarded to the scientist. This process is mirrored when the insurance company sends its encrypted data to the police for further encryption and routing to the scientist. The scientist now holds two encrypted sets of data representing crime and depression statistics but neither the police nor the insurance company can decrypt any of the data and the scientist is equally helpless. The encryption technique allows, however, that some aspects of the encrypted data sets are compared directly without decryption. The scientist can thus count the number of subjects that are both criminal and depressed without getting any information whatsoever about the crime or depression status of any of the subjects. Similarly, it is possible to count the total number of depressed people without learning anything about the psychological status of individual subjects.

3.4.2 Background

PPDDM-V algorithms consist of three hierarchical layers,

1) Encryption and coding techniques

2) Secure sub-protocols

3) Application algorithms

Each layer has its own research literature. This section describes encryption and coding techniques. It is assumed that the reader knows the basics about public key cryptosystems (Katz & Lindell, 2007) and

has studied the background section of the chapter on PPDDM-H and in particular homomorphic cryptosystems.

3.4.2.1 Commutative encryption

Consider a data item *v* that first is encrypted by a first party, the result is then encrypted by a second party, the new result is furthermore encrypted by a third party and so on. The multi-layer encrypted output is for three participants $R(v) = E_{pk(3)}(E_{pk(2)}(E_{pk(1)}(v)))$, where $E_{pk(i)}(.)$ means encryption by participant *i* using the key *pk(i).* If a *commutative encryption* algorithm is used the end result will be the same independently of the order of the encryption operations. Using a commutative algorithm ensures e.g. that $E_{pk(3)}(E_{pk(2)}(E_{pk(1)}(v))) = E_{pk(1)}(E_{pk(2)}(E_{pk(3)}(v))) = E_{pk(3)}(E_{pk(1)}(E_{pk(2)}(v)))$ for all *v* and all key values. It is further assured that $R(v)$ is unique so that $R(v_1) \neq R(v_2)$ if $v_1 \neq v_2$.

3.4.2.2 Run-length coding of a binary vector

A binary vector such as (0, 1, 0, 0, 1, 1, 0, 0, 0, 1) containing *k* ones can alternatively be represented by a *k*-dimensional vector where each component is the position index of a one in the binary representation. The run-length representation of our example vector is (2, 5, 6, 10). Run-length coding can be a compact representation of sparse binary vectors.

3.4.3 PPDDM-V association rule mining with commutative encryption

Many PPDDM-V applications use secure sub-protocols based on homomorphic encryption. Secure association rule mining algorithms can e.g. use secure homomorphic dot products for support calculations. The principles for this is, however, quite similar to the PPDDM-H methods that is described in the previous section. For educational purposes, we will therefore focus on a different approach to PPDDM-V that also has distinctive advantages compared to the homomorphic methods.

3.4.3.1 Objectives

PPDDM-V methods are needed in situations where multiple sites hold different attributes of the same transaction i.e. *vertical data partitioning*. Different authorities could e.g. compile different information about the same individual. The owners of the data sources are in general reluctant to share data but are willing to cooperate for the purpose of generating global statistics of common interest.

The objective of *PPDM* for distributed databases with vertical data partitioning (PPDDM-V) is the same as for PPDDM-H, namely to enable global data mining in distributed databases without trusted intermediaries and without disclosing any information to any of the participants beyond the intended result of the mining operation. PPDDM-V is also only concerned with what the participants can learn directly from performing the protocols and ignores the possible use of information resources beyond the focus of the protocols. The PPDDM-V methods that we have found in the literature handle only the Semi-Honest adversarial model where participants can be trusted to follow the agreed protocol.

3.4.3.2 PPDDM-V association rule mining

To exemplify state-of-the-art PPDDM-V methods, we consider an association rule mining situation with n items and m transactions (see section 2.2 for a brief introduction to association rule mining). The sales history of each item is represented by an m-dimensional binary attribute vector. Bit number p in the attribute vector of a given item is set to one if the item was sold in transaction number p and bit number p is set to zero otherwise. The set of attribute vectors are distributed among ℓ parties, each holding one or more attribute vectors. We can think of each party as the vendor of a subset of items. The key process in distributed association rule mining is to calculate the support for a set of items. Note that the support of a set of items X is the proportion of transactions where X is a subset of the items in the transaction record. To calculate the support of X we need to access the databases of all the vendors that provide items belonging to X. The PPDDM-V process for this is as follows.

1) One of the sites, say site 1, is selected to be the master site that collects the results. Several or all sites could alternatively take the role of master site.
2) All sites perform a run-length coding of their attribute vectors.
3) Each site *i* generates an encryption key *pk(i*).
4) The following process is performed for all components r_{ks} of all run-length coded attribute vectors $\mathbf{r}_k$:
    a. The owner of r_{ks} , site *i* encrypts r_{ks} using commutative encryption and sends the result $E_{pk(i)}(r_{ks})$ to site *i+1* .
    b. Each site $q \neq i$ that receives the encrypted r_{ks}, encrypts the input with commutative encryption using the key *pk(q*) and forwards the result to the site *q+1* or if *q*= ℓ to the site *q*=1.
    c. After a complete roundtrip, site *i* receives the fully encrypted result $R_{ks} = E_{pk(1)}(E_{pk(2)}(...E_{pk(\ell)}(r_{ks})...))$. Note that the encryption operations, using the properties of commutative encryption, have been put in a standard order.
    d. Site *i* sends R_{ks} to the master site.
5) The master site compiles completely encrypted versions $\mathbf{R}_k$ of all attribute vectors *k* where the components of are R_{ks}.
6) The master site computes the support of any item set *X* by selecting the set of $\mathbf{R}_k$ that corresponds to the items in *X* and counting the number of component values that are common to all the $\mathbf{R}_k$ in the set. Assume e.g. that the tenth transaction contains the itemset *X* . This means that the run-length coding of all the $\mathbf{r}_k$ that belong to the set always include some component r_{ks}=10 where the component index *s* typically will differ between the different run-length encoded attribute vectors. The master site knows only the completely encrypted representation R_{ks} but the commutative property of the encryption algorithm guarantees that there is a unique one-to-one mapping between r_{ks} and R_{ks}. The common cleartext component r_{ks}=10 will hence engender a common cryptotext component R_{ks}. Counting the number of common components in the encrypted set and dividing with the total number

of transactions gives the support for the selected data set. The process for calculating the confidence for an association rule uses a different itemset but is otherwise identical.

Note that the master site does not necessarily have a privileged position since all participants can receive the set of encrypted attribute vectors and thus verify the data mining operations. None of the participants learns anything else than the output of the support calculation. A particular advantage of this algorithm is that any number of support calculations can be performed once the expensive layered encryption operations have been performed.

3.4.4 Comments and conclusions

The PPDDM-V association rule mining method that is outlined in the previous section is in a malicious adversarial scenario vulnerable to probing attacks where a malicious participant submits false data that has been engineered to reveal sensitive aspects of other participant´s data. A simple version of this mode of attack is that the malefactor submits an attribute vector representing that only one transaction includes a sensitive item. If the secure association rule mining process returns a support count of one it is revealed that the target transaction includes all the items in the query. Repeated probing attacks would enable an aberrant participant to successively map other participant's secret data.

We have not found any research addressing the trade-off between privacy and utility or trade-off between privacy and efficiency in PPDDM-V methods beyond the obvious measure of restricting secure protocols to really sensitive data and to adapt the encryption key length. Computational complexity, mainly caused by encryption operations, is a very serious issue in PPDDM-V. The total number of encryption operations in the secure association rule mining method that is described in the previous section is of the order of magnitude of $n \cdot m \cdot k$ where n is the number of items, m is the number of transactions and k is the number of sites. The computational load is, however, well distributed so that each site performs $n \cdot m$ encryptions. Since a 512 bit encryption takes about 10 microseconds on a typical computer this means that even a quite moderate distributed database of

10.000 transactions and 100 items engenders an execution time of more than two hours at each site. Many association rule mining operations can, however, be performed once the database has been converted to the commutative encrypted form.

PPDDM-V methods is just like PPDDM-H methods best suited for applications where the participants are major organizations each controlling their own data and where data collectors and record owners trust or are forced to accept how the organizations handle the data.

3.5 Privacy-preserving methods for unstructured text

In the data privacy community as a whole, a considerable amount of research has been devoted to developing various privacy-preserving techniques, but as noted by Gardner & Xiong (2009), the focus is on structured data only. In this section we will briefly look at methods that can be applied to unstructured text.

3.5.1 Tutorial example

To illustrate the problem of anonymization of unstructured text, consider a class where medical students are supposed to learn from real medical records of patients suffering from various diseases. The medical records are written in free-text by doctors, and the goal is to preserve the patients' privacy by removing any information that can be used to identify the true identity of a patient (e.g., phone numbers, names, etc.). A perfect solution would remove or alter all such information and at the same time preserve as much of the text as possible so that all relevant information is kept in the documents.

3.5.2 Background

We have not found any systematic reviews of how to anonymize unstructured text in the general case. However, there are strong connections to the field of information extraction and, more specifically, the problem of named entity recognition (NER), in which the focus is to discover entity information in unstructured text. We describe a number of methods for finding occurrences of e.g. person

names, locations, and phone numbers in unstructured documents. We assume the reader is having a basic understanding of statistical learning approaches to text mining, such as naïve Bayes classifiers or support vector machines (SVMs), as well as pattern matching techniques. In addition to the more general literature on NER, there also exists a significant amount of work on how such techniques can be applied for de-identification (scrubbing) of medical records. Those methods are typically targeted at removing certain types of data elements (e.g., dates, locations, and phone numbers) referred to as Protected Health Information (PHI), since removal of PHI is a way to fulfil acceptable de-identification of clinical records in the United States, as defined by the Health Insurance and Accountability Act (HIIPA). Even if de-identification of a text is successful, this does not mean that all information that can be used to identify a certain individual has been removed. De-identification only means that explicit identifiers are removed (or replaced), while anonymization implies that it should not be possible to link the data to an individual (Meystre et al., 2010).

3.5.3 Method overview

Automated text de-identification applications, as well as more general NER-applications are most often based on pattern matching techniques, machine learning techniques, or a hybrid combination of the two. An overview of recent research on automatic de-identification of textual electronic health records is presented in (Meystre et al., 2010). Many of the applications presented there are also addressed in further detail in (Uzuner et al., 2007), in which the results from a challenge on removing PHI from medical discharge records are described. The same kind of techniques can be identified when reviewing more general NER applications. Most general NER systems have mainly been focused and tested on recognizing organizations, persons, locations, dates, times, etc. in newswire text. A brief overview of NER research can be found in (Nadeau et al., 2007). The interested reader is encouraged to use these references for further reading but the most important findings from the surveys are summarized below.

Many traditional approaches to NER use dictionaries or *gazetteers* containing common person, organization, and location names. Such

dictionaries work well for common names and places, but it has been known for a long time that they are not sufficient for very unusual names or misspellings (they will not be in the list), and more problematic, will result in ambiguities in many cases (e.g., due to lexical overlap between PHI and non-PHI). Dictionaries are together with rules and regular expressions often used in methods relying on pattern matching. A problem with all such approaches is that they typically are manually constructed, demanding a lot of expert knowledge and time. Moreover, they have a limited generalizability. However, there are also many advantages. Compared to the machine learning algorithms presented below, a clear benefit with pattern matching techniques is that they do not require any labelled (annotated) training data. It is also fairly straight-forward to add rules or dictionary entries to improve the performance.

More recent applications to NER and de-identification seems to be more and more focused on machine learning solutions, or hybrids between machine learning and pattern matching techniques. Such methods include conditional random fields (CRFs), decision trees, maximum entropy, naïve Bayes, and support vector machines (SVMs), where SVMs seem to be most heavily used in most applications. The main advantage of such methods is that they are able to automatically learn complex patterns and that limited domain knowledge is required for the developers. The main drawback is the large amounts of labelled data required for the learning. Another problem is that many of the methods are "black boxes" that are hard to interpret and correct in case there are classification errors.

The features used for classification vary between implementations and systems, but examples of features that are used are: the category of a sentence (as determined by a sentence classifier), part-of-speech tags, special characters (e.g., capital letters), the position of a sentence in a record, token length, length of sentence, and various format patterns.

The results presented by Uzuner et al. (2007) suggest that the best available techniques for de-identification are able to find almost all instances of PHI (achieving precision higher than 95%) and use a combination of machine learning and pattern matching algorithms. However, as noted in the same paper, it is hard to know how well the used systems would generalize to data from another medical domain

(or even more problematic, an entirely different domain). To remove PHI correctly is also only one side of the coin. It is also important not to over-scrub the data, i.e., to erroneously remove non-PHI data. If over-scrubbing was not of any concern, all data could be removed completely. In most cases it is probably much worse to classify PHI as non-PHI than classifying non-PHI as PHI, but it is hard to judge what balance that is the best.

3.5.4 Comments and conclusions

Although the current systems for medical de-identification are not fully perfect, it is worth noticing that also humans perform errors on the de-identification task (as shown in (Sweeney, 1996)). For this reason, one could argue that it is at least as safe to use computers as humans for de-identification purposes. The problem is that is hard to know who to blame if an automatic de-identification system fails and a patient's identity is revealed based on PHI that should have been removed. Moreover, methods suggested for the medical domain are at best "guaranteeing" de-identification of PHI, but as should be obvious by now, this is not the same thing as guaranteeing anonymization. Even though all PHI is removed from a document, this is not a guarantee for that individuals cannot be identified through combinations of various quasi-identifiers remaining in the text. While there has been quite a lot of research on both data privacy in structured text and de-identification of medical records consisting of unstructured text, it is worth noticing that very little has been done on the intersection of those. As argued by Gardner and Xiong (2009), "efforts on de-identifying medical text documents in medical informatics community rely on simple identifier removal or grouping techniques without taking advantage of the research developments in the data privacy community". An interesting first step to bridge the gap between the two problems is presented in (Gardner and Xiong, 2009).

Closely related to the problems of de-identification and anonymization is the problem of document sanitization. This can be defined as the removal of sensitive information from a document with the purpose of reducing a document's classification level. Not much work seems to have been done on automatic document sanitization, but it is yet

another application area for anonymization techniques for unstructured text.

According to Uzuner et al. (2007), a currently popular approach is to release data for research purposes by forcing the recipient to agree contractually not to try to re-identify patients. In this way, a juridical rather than a technical approach is used to solve the problem.

3.6 PPDM for network data

Data mining in network data including in particular social networks raises many privacy concerns. In a social network vertices correspond to individual users and links represent friendship connections. Vertices and links may have attributes that further characterize the user and the nature of the connections between users. Adversaries may be interested in exploring sensitive data by mapping individuals to vertices, revealing friendship relations and learning the attributes of vertices and links. Statistical properties such as an individual's number of friends and relations to groups could also be sensitive.

Naïve anonymization means that all explicit identifiers including in particular vertex identifiers are replaced by dummy identifiers such as randomly selected integers. This gives good protection if adversaries have no background information but can be quite brittle if background data on similar networks are available.

Using the Friendster network of 4.5 million nodes, Hay et al. (2008) found that the local network structure is quite revealing. Hay et al. exploited that the background network often has the same local structure as the network under attack and that users are explicitly identified in the background network and found that about 50% of the users could be identified if the structure of the local network is known at up to two levels of neighbours. Adversaries may also know the identity of some seed nodes in both the target and the background network. Narayanan and Shmatikov (2009) used 150 seed nodes with anonymized Twitter as the target and Flickr as the background network finding that 31 % of 30 000 overlapping individuals could be correctly identified while 12 % were incorrectly identified.

Given the relative ease of node re-identification in anonymizised networks there is a need for PPDM methods for networked data. One

approach is to dynamically filter responses to queries about network information for the purpose of thwarting attempts to map large portions of target networks. Network data could alternatively be sanitized before publishing a modified version of the network. Sanitation methods can broadly be divided in graph-modification and graph-clustering methods. Graph modification means that vertices and connections are changed according to stochastic or deterministic algorithms for the purpose of precluding re-identification by reducing the similarity between the sanitized target network and background data. Graph clustering involves grouping of vertices or edges before publication.

As an example of a PPDM-method of the former type we shall dwell on k-degree anonymization (Liu & Terzi, 2008). The degree of a node is the number of connections to other nodes. Since vertices potentially can be re-identified based on the degree, k-degree anonymization ensures that every node has the same degree as at least *k*-1 other nodes in the sanitized network. This is achieved by judiciously adding or deleting connections for the purpose of realizing k-degree anonymization while simultaneously optimizing a utility objective. Liu and Terzy (2008) suggest that the symmetric difference of the sets of edges in the original and the sanitized networks respectively is minimized. The symmetric difference between two sets is the number of elements that is unique for one of the sets.

Apart from the algorithmic complexity of the optimization process, k-degree anonymization suffers from a privacy problem that is generic to all PPDM methods where data is modified by algorithms that ignores semantic aspects. Suppose that a friendship link to a known serial murderer inadvertently is fabricated for the purpose of achieving k-degree anonymization. A re-identification attempt could then indicate that a blameless user has befriended an infamous felon. The indication is not certain but have a low but significant probability. This might be enough to cause that innocent people are harassed by the press or even arraigned to courts. Algorithmically correct but semantically insensitive sanitation could hence have a very serious impact on privacy.

3.7 Other methods

This section briefly describes PPDM methods that are interesting from a technical point of view but presently are somewhat peripheral in the research literature.

3.7.1 Importance weighting

Elkan (2009) introduces a new PPDM method that is based on the existence of a public dataset *E* that has records of the same data types as the secret dataset *D*. For simplicity we assume that all records are real-valued numbers. It is further assumed that *D* and *E* are drawn from the probability distributions *f* and *g* respectively. Egan notes that the average of any function *b* can be computed according to the *importance sampling identity*,

$$\int b(x) f(x) dx = \int b(x) w(x) g(x) dx ,$$

where the weight function is $w(x) = f(x) / g(x)$. This means that by knowing the secret data set $D = \{x_1, x_2, ...x_n\}$ and the public data set $E = \{z_1, z_2, ...z_n\}$, it is possible to calculate a set of weights $\{w_1, w_2, ...w_n\}$ so that averages of any function *b* can be estimated according to,

$$\frac{1}{n}\sum_{i=1}^{n} b(x_i) = \frac{1}{n}\sum_{i=1}^{n} b(z_i) w_i$$

Publishing the weight vector and a pointer to the public data set will hence allow anyone to estimate averages of the secret data set.

3.7.2 Classifier downgrading

The classifier downgrading method (Chang & Moskowitz, 2000; Chang et al., 1998) is a systematic approach to removing attributes in data for the purpose of enhancing privacy and is therefore similar in purpose to k-anonymity. Consider a data set where each record includes an explicit identifier of the subject. A multiclass classifier is trained for the purpose of predicting the explicit identifier from the

remaining attributes. If the classifier is able to predict the value of an explicit identifier one of the attributes that contribute to the successful classification is removed and replaced with a wild card symbol. Which attribute to remove is selected by computing the mutual information between the attributes and the explicit identifier and removing the attribute with the highest mutual information. After removing the attribute, the classifier is retrained on the downgraded data and the process is repeated until the classifier fails to predict the explicit identifier. This failure means that there are no quasi-identifiers left in the downgraded data, at least not from the point of view of the classifier.

It is, however, known that no classification algorithm is superior for all possible data sets (the no free lunch theorem). This means that it is possible that a different classifier will be able to predict the explicit attributes thus pointing to quasi-identifiers that are hidden from the first classifier. The downgrading process is also very expensive computationally and hence not realistic to apply in its present form. Classifier downgrading provides, however, conceptually an interesting alternative to k-anonymity.

4 Conclusions

4.1 Applications of PPDM

We have not found any examples of real-life use of any of algorithms that has been suggested in the recent spate of research interest. The PPDM literature mentions a few success cases but they are based on an older generation of methods. The Scrub system from 1996 is a specialized application for anonymization of partially handwritten medical records (Sweeney, 1996). The Datafly system from 1997 (Sweeney, 1997) works by limiting the size of database fields containing sensitive attributes of medical records. This approach is similar to the common practice of showing just a few figures of a credit card number on receipts.

Many application areas of PPDM have been suggested, including the medical domain, social security, police and surveillance, genetic and forensic data, and business intelligence. While it is quite likely that PPDM techniques eventually will be applied in all of these areas it is also important to note the factors that are checking the advance of PPDM.

Privacy protection is in many domains governed by legal requirements that have no provisions for graded privacy preservation according to for example the randomization or k-anonymity approaches. Database owners are often more concerned with avoiding legal complications than with the utility of the published data and are therefore reluctant to take the risk of employing a technique that is difficult to explain to authorities and to the general public. If PPDM methods become established in one or several bridgehead domains it is possible that laws or canonical interpretations of laws adapt to allow graded privacy preservation techniques.

Data that is privately owned and not subject to legal privacy requirements is often mentioned as a promising first bridgehead of the PPDM technology. Such data could for example be sales information that corporations want to share in a controlled manner for the purpose of generating statistics of common interest. While PPDM certainly is applicable in this context, it should be noted that the business

managers typically are unaware of the PPDM research and that traditional solutions such as providing the data to a trusted mediator or sharing the data under the protection of legal agreements might be considered to be safer or simpler solutions. A prerequisite for large-scale introduction of PPDM in the business sector is that a major provider of business intelligence software provides PPDM as a part of an integrated solution.

In the military and security sector PPDM methods could be used for sharing operational and intelligence data for example within loose international coalitions. The motivations and hurdles for the introduction of PPDM techniques are similar to those of the business sector. Armed forces are unimpeded by privacy law for at least some databases. There are incentives for sharing data but also reasons against unlimited sharing. The current PPDM research offers a cornucopia of concepts for balancing mutual utility with privacy concerns. Users of PPDM services in the military and security domain would, however, need military-grade PPDM products integrated in whole solutions and delivered by a creditable contractor.

PPDM for distributed databases with military-grade cryptographic methods and secured for adversarial attacks could become the first real-live PPDM application in the military sector. The main problem with this technique is computational complexity. This might, however, not be a show-stopper for applications where a small amount of high-value data is exchanged between military forces.

Sanitation-based PPDM should be applied to sensitive intelligence data with great caution since the curse of dimensionality and the general brittleness with respect to adversarial attacks imply that the risk for unwanted re-identification is considerable.

4.2 The state of PPDM

Much of the present generation of research PPDM methods are not ready for large-scale application. Computational complexity, the curse-of-dimensionality effect and unaccounted attack modes are generic remaining problems that impact on different methods with various degree and severity. A second generation of methods is needed where computationally feasible approximate techniques with well understood

limitations and performance are developed. Successful methods must also be implemented in commercially available products delivered by mainstream system integrators. Business and legal practice may have to adapt to the new technical opportunities.

In spite of the weaknesses of current PPDM methods they might be useful in situations where publication of potentially sensitive databases is unavoidable for example because of legal requirements. Manual anonymization, is as noted in section 3.6, known to be fault-ridden so even imperfect PPDM might improve performance or reduce costs.

It should also be noted that privacy-preservation can be generalized beyond the application to protecting individuals. In the context of military intelligence PPDM methods could for example be applied to the task of hiding the identity of military units or the nationality of unmanned systems.

5 References

Agrawal, D. & Aggarwal, C. C. (2001), On the Design and Quantification of Privacy Preserving Data Mining Algorithms, in 'Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'01)', pp. 247-255.

Agrawal, R. & Srikant, R. (2000), 'Privacy-Preserving Data Mining', ACM SIGMOD Record 29(2), 439-450.

Bayardo, R. J. & Agrawal, R. (2005), Data Privacy through Optimal k-Anonymization, in 'Proceedings of the 21st International Conference on Data Engineering', IEEE Computer Society, Washington, DC, USA, pp. 217-228.

Carlisle, D.; Rodrian, M. & Diamond, D. (2007), 'California inpatient data reporting manual,medical information reporting for California', Technical report, Office of Statewide Health Planning and Development.

Chang, L., M. I. (2000), A Decision Theoretical Based System for Information Downgrading, in 'Joint conference on information sciences'.

Chang, L. & Moskowitz, I. S. (1998), Parsimonious downgrading and decision trees applied to the inference problem, in 'Proceedings of the 1998 workshop on new security paradigms', ACM, New York, NY, USA, pp. 82-89.

Chen, K.; Sun, G. & Liu, L. (2007), Towards Attack-Resilient Geometric Data Perturbation, in 'Proceedings of the 2007 SIAM International Conference on Data Mining', pp. 78-89.

Elkan, C. (2010), Preserving privacy in data mining via importance weighting, in Aikaterini Mitrokotsa Vassilios S. Verykios Yücel Saygin Christos Dimitrakakis, Aris Gkoulalas-Divanis, ed., 'Privacy and Security Issues in Data Mining and Machine Learning'.

Feigenbaum, J.; Ishai, Y.; Malkin, T.; Nissim, K.; Strauss, M. J. & Wright, R. N. (2006), 'Secure multiparty computation of approximations', ACM Transactions on Algorithms 2(3), 435-472.

Gardner, J. & Xiong, L. (2009), 'An integrated framework for de-identifying unstructured medical data', Data Knowl. Eng. 68(12).

Hay, M.; Miklau, G.; Jensen, D.; Towsley, D.& Weis P. (2008), 'Resisting structural re-identification in anonymized social networks'', Proceedings of the VLDB Endowment, vol. 1, pp.102-114.

Jakobsson, M.; Juels, A. & Rivest, R. L. (2002), Making mix nets robust for electronic voting by randomizedpartial checking, in 'Proceedings of the 11th USENIX Security Symposium', pp. 339–353.

Johnson, W. B. & Lindenstrauss, J. (1984), 'Extensions of Lipschitz mappings into a Hilbert space', Contemporary Mathematics 26, 189-206.

Katz, J. & Lindell, Y. (2007), Introduction to Modern Cryptography, Chapman and Hall/CRC Press.

Liu, K.; Kargupta, H. & Ryan, J. (2006), 'Random Projection-Based Multiplicative Data Perturbation for Privacy Preserving Distributed Data Mining', IEEE Transactions on Knowledge and Data Engineering 18(1), 92-106.

Liu, K. & Terzi, E. (2008), Towards identity anonymization on graphs, in 'ACM Special Interest Group for the Management of Data (SIGMOD)'.

Meyerson, A. & Williams, R. (2004), On the complexity of optimal K-anonymity, in 'Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems', pp. 223-228.

Meystre, S. M.; Friedlin, F. J.; South, B. R.; Shen, S. & Samore, M. H. (2010), 'Automatic de-identification of textual documents in the electronic health record: a review of recent research.', BMC medical research methodology 10(1).

Muralidhar, K.; Parsa, R. & Sarathy, R. (1999), 'A General Additive Data Perturbation Method for Database Security', Management Science 45(10), 1399-1415.

Nadeau, D. & Sekine, S. (2007), 'A survey of named entity recognition and classification', Linguisticae Investigationes 30(1), 3-26.

Narayanan, A. & Shmatikov, V. (2009), 'De-anonymizing social networks', Security and Privacy.

Sweeney, L. (1997), Guaranteeing anonymity when sharing medical data, the Datafly System, in 'Proc AMIA Annual Fall Symposium'.

Sweeney, L. (1996), 'Replacing personally-identifying information in medical records, the Scrub system', Journal of the American Medical Informatics Association, 333-337.

Uzuner, O.; Luo, Y. & Szolovits, P. (2007), 'Evaluating the state-of-the-art in automatic de-identification.', Journal of the American Medical Informatics Association : JAMIA 14(5), 550-563.

Yang, Z.; Zhong, S. & Wright, R. N. (2005), Anonymity-preserving data collection, in 'Proceedings of the11th ACM SIGKDD Conference', ACM, , pp. 334-343.

Aggarwal, C. C. & Yu, P. S., ed. (2008), ‘Privacy-Preserving Data Mining - Models and Algorithms’, Vol. 34, Springer.

Fung, B. C., ed. (2011), 'Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques', Chapman & Hall.