Integral Privacy

Vicenç Torra[1] Guillermo Navarro-Arribas[2]

[1] School of Informatics,
University of Skövde, Sweden
[2] Department of Information and Communication Engineering,
Universitat Autònoma de Barcelona, Catalonia, Spain
Email: vtorra@his.se, guillermo.navarro@uab.cat

Abstract. When considering data provenance some problems arise from the need to safely handle provenance related functionality. If some modifications have to be performed in a data set due to provenance related requirements, e.g. remove data from a given user or source, this will affect not only the data itself but also all related models and aggregated information obtained from the data. This is specially aggravated when the data are protected using a privacy method (e.g. masking method), since modification in the data and the model can leak information originally protected by the privacy method. To be able to evaluate privacy related problems in data provenance we introduce the notion of integral privacy as compared to the well known definition of differential privacy.

1 Introduction

Data provenance permits to track where data come from and how these data have been combined in order to produce new data elements. Data provenance is used to improve data quality, and have been used in a quite number of different areas including scientific data, e-science, accounting (financial data), and medical data [3, 9, 1].

Data privacy is the area that studies methods and techniques to avoid the involuntary release of sensitive data [5, 11, 10]. Methods are used because of companies own interest to keep their information private, but also because of existing regulations. In 2016 the new EU General Data Protection Regulation was entered into force, a regulation that shall apply from 25 May 2018. This regulation consolidates two rights: the right to be forgotten and the right to amend.

Companies need appropriate software so that they can guarantee these two rights to their customers. Note that the right to be forgotten does not only imply that customers can force the deletion of records with their data, but also that aggregated data and inferences extracted from their data need to be reconsidered and eventually modified or also deleted.

Data provenance has a tight relation with data privacy. On the one hand, data provenance is essential to implement these two rights. We need to keep track of how data is processed and aggregated in order to know what needs to be deleted, amended or reconsidered when records are deleted or amended. Otherwise, we will need to delete all what follows from a record once there is a requirement to delete such record.

On the other hand, data provenance poses specific questions to data privacy. Note that provenance information may be confidential, provenance information cannot be modified at will, etc. See e.g. [7, 2], for a review of problems and solutions related to data provenance and data privacy.

In this paper we discuss privacy models. We present a new privacy model insipired on data provenance, on the two mentioned rights, and how all these aspects relate to data privacy.

We call this model integral privacy, to compare it with differential privacy [6]. As we will see later, while differential privacy focus on the *output* of a function from the data (a computation), this model focus on the *input*. While differential privacy computes differences between outputs, here we consider a set of modifications of the input.

The structure of the paper is as follows. In Section 2 we review the notation we use in the paper. In Section 3 we present our definition and in Section 4 we compare integral privacy with differential privacy. The paper finishes with a summary and a discussion of future work.

2 Notation and problem set up

We will consider a set X (a file or a database) to which we have applied some modifications μ to reach a data set X'. We will denote the fact that X' is constructed from X with some modifications μ by the expression $X' = X + \mu$.

Then, using algorithm A we extract knowledge G and G' from X and X', respectively. If we apply a masking method ρ to X and X' we get χ and χ' from which we obtain knowledge Γ and Γ' using algorithm A. Figure 1 represents these data sets, methods and algorithms. This conforms the full picture of our scenario. Provenance data are included in all the data sets and the figure shows all possible cases one can find in processing the original data set X.

$$\begin{tikzcd}
X \arrow[rrr, "\mu"] \arrow[dr, "A"] \arrow[dd, "\rho"'] & & & X' \arrow[dr, "A"] \arrow[dd, "\rho"'] & \\
& G & & & G' \\
\chi \arrow[dr, "A"] & & & \chi' \arrow[dr, "A"] & \\
& \Gamma & & & \Gamma'
\end{tikzcd}$$

Fig. 1. Original file X with protected file χ and knowledge/models G and Γ extracted from X and χ, respectively. Updated file X' and protected file χ' with knowledge/models G' and Γ' extracted from X and χ', respectively. Protection method ρ and knowledge discovery algorithm A.

3 Integral privacy

In this section we propose our definition for privacy. It focuses on the modification μ that apply to the original dataset X following the notation introduced in Section 2. We make explicit our assumptions on what an intruder may know. We then state the intruders goal. We consider that the intruder can be a person that is working outside the data holder (the company with the database X) or an insider with partial access to the data and the knowledge extracted (either from the database or possibly also using some information obtained from other sources).

3.1 Specific scenario #1

To introduce the notion of integral privacy we first consider an scenario where the intruder knows: $S \subset X$, G, G'. That is, the intruder has partial knowledge of the data in the database (the worst case scenario is when $S = X$, the best case scenario is when $S = \emptyset$).

The privacy requirements are that intruders cannot be able to determine μ and $S' \subseteq X \setminus S$ with certainty. That is, that the intruder cannot find neither records from the file, nor information about the modifications.

3.2 Intruder's goal

The main goal of the intruder can be summarized as follows. Given $S \subset X$, G, G', find the set of possible modifications μ that are consistent with data $S \subseteq X$ and knowledge G and G', and find elements in $X \setminus S$. Under the transparency principle, we may assume that the intruder knows the algorithm A used to generate G.

We illustrate this problem with an example. The example uses ID3, one of the simplest decision tree learning algorithms for categorical data and with no pruning. In the worst case scenario (i.e., when $S = X$), and assuming that G is obtained by means of the application of the ID3 algorithm to X, this problem is to find the modifications μ such that $G = ID3(X)$ and $G' = ID3(X + \mu)$. In the general setting, the problem is to find the following set of modifications, for a given algorithm A

$$\mathscr{M} = \{\mu | G = A(X) \text{ and } G' = A(X + \mu)\}.$$

On the set of modifications For a large number of machine learning algorithms, the set of modifications $\mathscr{M}$ is not a singleton. To support this statement, let us consider Gen and Gen' the set of generators of G and G', respectively. That is, the set of data that lead to G and G' when we apply to them the algorithm A. Then, note that when there are several generators Gen and Gen', the set of possible transformations μ is not a singleton. Note that

$$\cup_{g \in Gen, g \in Gen'} \{g' - g\} \subseteq \mathscr{M}.$$

Now we consider a few cases in which algorithms ensure that a model has different generators. In all these cases, due to the result above, we will have sets $\mathscr{M}$ that are not a singleton.

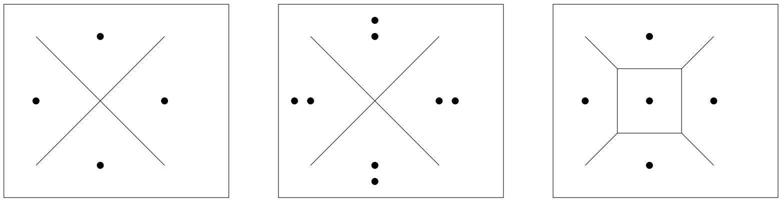

Fig. 2. Three Voronoi maps. The first one (left) containing only open regions, the second one (center) with the same regions but with the original generators and a new set of generators. The third one (right) with a closed region.

We first consider that A is the algorithm for 1-nearest neighbor. It is known that the model built can be represented with a Voronoi tesselation. Let X be defined in a domain D. When all regions are open (i.e., as in Figure 2 (left)), then we can construct sets $\hat{X}$ with $\hat{X} \cap X = \emptyset$ and such that generate the same map. They consist of displacing the points in X out of the map. See Figure 2 (center). When there are closed regions (i.e., as in Figure 2 (right)), the points of closed regions cannot be changed. So, in case that another set $\hat{X}$ can generate the same map, there will be points that cannot be changed $\hat{X} \cap X \neq \emptyset$.

In these constructions, we were considering that X and $\hat{X}$ had the same number of points (records). We will consider a more general case now in which X and $\hat{X}$ have a different number of records. This causes that the model from X and the model from $\hat{X}$ have a different number of regions.

In a classification problem, what is rellevant for our model is the class associated to each element. In the case of Voronoi tesselations for a 1-nearest neighbor this can be modeled with colors (or assignments) to each region. Let $G_c(p)$ be the color assigned to position p in the map. We say that two Voronoi tesselations G and G' are color-equivalent if $G_c(p) = G'_c(p)$ for all p even in the case that the number of regions is different.

Let us consider the case of a Voronoi tesselation in which the color of adjacent regions is all different. Then, the question is whether there exist a set $\hat{X}$ (with more records than X) such that the Voronoi tesselation generated from $\hat{X}$ is color-equivalent to the one in X.

Let x be the points in X. Let a, c be two points of X such that they generate a border in G. Let z be the point, $z = (a+c)/2$. Then, the points $a_c = (a+z)/2$ and the point $c_a = (c+z)/2$ are included in $\hat{X}$.

We can prove that all p that are at the same distance from a and c, they are also at the same distance from a_c and c_a. This can be proven as the two right triangles defined by the points (a_c, z, p) and (c_a, z, p) have two sides with the same length. So, the third should also have the same length. This implies that, at least for some examples, the border of the regions we have in X are also border of regions in $\hat{X}$. In such cases the procedure results into another set $\hat{X}$ (with a different number of elements) that represents the same map. That is, the model built from X and $\hat{X}$ is the same: $G = A(X) = \hat{G} = A(\hat{X})$.

Decision tree learning returns a decision tree from a data set. In the case of ID3, the tree is built for categorical data recursively selecting at each point the attribute that maximizes the information gain (or minimizes the entropy). Data sets that lead to the same entropy will produce the same trees. Nevertheless, even in the case of different entropies, the trees will be the same if the set of attributes that maximize the entropy are the same.

For any linear regression model, the number of sets that can generate the model is infinite. However, when constraints exist for the generators (e.g. integer data in a given domain) this may not be the case.

We have shown that when different datasets can generate the same knowledge, $\mathscr{M}$ is not a singleton. In addition to that, for some algorithms, when μ is a set of valid modifications, then there is another set $\mu \subseteq \mu'$ that is also a valid set of modifications. The following example illustrates this case.

Example 1. Let X be a set of n records where $n-1$ of them are of class $+$ and 1 is of class $-$. Then, let $G = A(X)$ be a decision tree with two branches and one question. Let $G' = A(X')$ be a decision tree with a single node and no question assigning always the class $+$. Then, it is clear that all modifications in $\mathscr{M}$ include the deletion of the record in class $-$.

Therefore, if μ corresponds to the deletion of the record in class $-$ and μ' are all other possible modifications, then μ' includes μ. In this framework, we can consider the set (or sets) of possible transformations, and the lattice defined from this set of transformations and the subset inclusion. Note that it is also rellevant to consider the intersection of all $m \in \mathscr{M}$. In the example, this intersection corresponds to the deletion of the record of class $-$. Similarly, it is relevant to consider the minimal elements of the lattice. That is, the modifications that are minimal with respect to the set inclusion. The minimal modifications are rellevant for an intruder.

We finish this discussion with the following remarks.

- When we only allow deletions, the number of modifications is finite (for a finite database). Therefore, the set of minimal modifications is also finite.
- The set of generators of a real data set is smaller than the set of possible generators. In real applications, not all modifications are possible, and not all possible modifications are equally plausible.

3.3 Privacy problem

In order to take into account the intruder goal described in Section 3.2 we consider the following privacy problem.

Find algorithms A that maximize the uncertainty of the intruder (with respect to the set of possible modifications). That is, we are interested in machine learning methods A such that the set

$$\mathscr{M} = \{\mu | G = A(X) \text{ and } G' = A(X+\mu)\}. \tag{1}$$

is large, and such that

$$\cap_{m \in \mathscr{M}} m = \emptyset. \tag{2}$$

The rational of this definition is that intruders cannot use their knowledge on the set of possible modifications to infer that a particular modification has taken place. The larger the set of modifications, the larger the uncertainty of the intruder. In addition, we do not want that even in the case of a large set, all modifications agree on a small set. This is to avoid situations as the one in Example 1.

3.4 Integral privacy definitions

On the basis of the previous discussion we introduce some definitions for privacy.

We define *i*-integral privacy when $\mathscr{M}$ defined according to Equation 1 is *large* and such that the intersection in Equation 2 is empty.

We define integral privacy à la k-anonymity, when the set $\mathscr{M}$ contains at least k alternatives.

We define k-anonymous integral privacy when the set $\mathscr{M}$ has at least k minimal elements.

With these definitions, we can consider solving the privacy problem above (for integral privacy) combining machine learning algorithms with data privacy algorithms. In this case, we define $\hat{A}(X) = A(\rho(X))$. Then, the scenario is similar to the one above but permits us to find good masking methods for a given algorithm A. The formulation is as follows.

Given X, G, G', and an algorithm A, a good masking method ρ is the one that makes the set

$$\mathscr{M} = \{\mu | G = A(\rho(X)) and G' = A(\rho(X+\mu))\}$$

large and such that $\cap_{m \in \mathscr{M}} m = \emptyset$.

We can consider additional restrictions for the set $\mathscr{M}$ as above.

3.5 Other specific scenarios

Section 3.1 introduced the main scenario that motivates the definition of integral privacy, but one can find other cases and possible scenarios. Here we provide a brief description of 4 more cases that can arise from the main problem description from Section 2.

- **Scenario #2.** Known by the intruder: χ, χ'. Intruders should not determine neither $S \subseteq X$ nor μ with certainty. That is, the intruder cannot find neither records from the file, nor information about the modifications.
- **Scenario #3.** Known by the intruder: X', G, G'. Similar to the first scenario from Section 3.1 but with X' instead of X.
- **Scenario #4 and #5.** Similar to cases #1 and #3 but knowledge is generated from $\rho(X)$. That is, we are considering Γ and Γ'. Under the transparency principle, we can also presume that the intruder is aware of methods ρ and A.

These three scenarios complement the one introduced previously and can contribute with more examples of the utility of our definition of integral privacy. It is important to note that some of these scenarios are equivalent to already existing problems in data privacy. For instance, scenario #2 can be considered as the problem of publishing protected dynamic data. Note also that when in scenario #1 we have that the algorithm A is a masking method ρ, it can be seen as equivalent to the second scenario.

4 Integral privacy and differential privacy

Our model can be considered as related to differential privacy. Nevertheless, the focus of our model differs to the focus in differential privacy.

In differential privacy, the main issue is to compute a query in a way that the output is insensitive to addition (or removal) of a single element of the database. This is achieved considering this computation as randomized and requiring that the distributions of the two outputs (the output of the computation on the original data set and the one of adding an element to it) are approximately the same. That is, for all X and x,

$$Distr(G(X)) \sim Distr(G(X+x)).$$

Note that this is for all databases X and for all possible elements that can be added into a database. Algorithms exist for achieving this goal, although for some type of data the noise required to ensure enough similarity may be very large. See e.g. [8].

Let us consider this problem from a different perspective. Let us assume that we know $G(X)$ and $G(X+x)$ (or their distribution) and that we know X. Consider for example the case of applying a decision tree learning algorithm to a data set. So $G(X)$ is a decision tree obtained from X using the algorithm. Then, $G(X+x)$ is also a decision tree. It can be the case that this other decision tree is quite different to $G(X)$ but that the set of possible records x that have generated $G(X+x)$ is very large.

For example, let X be a set with all records in the same class $+$ and then any record in class $-$ expands the tree. Alternatively, let $X+\mu$ be a set of records with classes $+$ and $-$ but with most of the records in $+$ and only a few in $-$ (so few that the deletion of one by a decision tree learning with pruning removes the $-$ class). For this example, we can consider that privacy is guaranteed at an appropriate level. Note that, in general, differential privacy (on $G(X)$ vs. $G(X+x)$) would not consider the process safe.

If we are interested in both types of privacy, we can define the concept of differintegral privacy that forces the data to satisfy differential privacy and integral privacy at appropriate levels. The term differintegral is borrowed from fractional calculus [4].

5 Conclusions

In this paper we have introduced the definition of integral privacy. The main goal is to provide tools for researchers to study data privacy when provenance data is present. We have provided a motivating scenario that yields the concept of integral privacy. This definition can be further developed in future works to comprise a framework for evaluating privacy in data provenance. Further work is needed to compute the set of modifications in different scenarios. This will permit us to evaluate methods with respect to disclosure risk and utility. Another line of future research corresponds to the case when instead of a single method A for extracting knowledge, we apply several of them $A_1, A_2, \ldots, A_n$ and thus we need to consider G_1 and G'_1, G_2 and G'_2, $\ldots$, G_n and G'_n.

Acknowledgements

Partial support by the Spanish MINECO (project TIN2014-55243-P) and Catalan AGAUR (2014-SGR-691) is acknowledged.

References

1. Barbier, G., Feng, Z., Gundecha, P., Liu, H. (2013) Provenance data in social media, Morgan & Claypool Publishers.
2. Bertino, E., Ghinita, G., Kantarcioglu, M., Nguyen, D., Park, J., Sandhu, R., Sultana, S., Thuraisingham, B., Xu, S., (2014). A roadmap for privacy-enhanced secure data provenance. J Intell Inf Syst 43, 481-501.
3. Buneman, P., Khanna, S., Wang-Chiew, T. (2001). Why and where: A characterization of data provenance. In International conference on database theory, pp. 316-330. Springer.
4. Das, S. (2008) Functional fractional calculus, Springer.
5. Domingo-Ferrer, J., Torra, V. (2001) A quantitative comparison of disclosure control methods for microdata, in P. Doyle, J. I. Lane, J. J. M. Theeuwes, L. Zayatz (eds.) Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies, North-Holland, 111-134.
6. Dwork, C. (2006) Differential privacy, Proc. ICALP 2006, Lecture Notes in Computer Science 4052, pp. 1-12.
7. Hasan, R., Sion, R., Winslett, M. (2007) Introducing secure provenance: problems and challenges, Proc. StorageSST, ACM, 2007.
8. Muralidhar, K., Sarathy, R. (2008) Generating Sufficiency-based Non-Synthetic Perturbed Data, Transactions on Data Privacy 1:1 17 - 33
9. Simmhan, Y. L., Plale, B., Gannon, D. (2005). A survey of data provenance in e-science. ACM Sigmod Record, 34(3), 31-36.
10. Torra, V., Navarro-Arribas, G. (2014) Data Privacy, WIREs Data Mining and Knowledge Discovery 4:4 269-280.
11. Winkler, W. E. (2004) Re-identification methods for masked microdata, Proc. PSD 2004, Lecture Notes in Computer Science 3050 216-230.