Explaining Recurrent Machine Learning Models: Integral Privacy Revisited

Vicenç Torra[1,2(✉)], Guillermo Navarro-Arribas[3], and Edgar Galván[4]

[1] Department of Computing Science, Umeå University, Umeå, Sweden
vtorra@ieee.org
[2] School of Informatics, Skövde University, Skövde, Sweden
[3] Department Information and Communications Engineering – CYBERCAT, Universitat Autònoma de Barcelona, Bellaterra, Catalonia, Spain
guillermo.navarro@uab.cat
[4] Naturally Inspired Computation Research Group, Department of Computer Science, Maynooth University, Maynooth, Ireland
Edgar.Galvan@mu.ie

Abstract. We have recently introduced a privacy model for statistical and machine learning models called integral privacy. A model extracted from a database or, in general, the output of a function satisfies integral privacy when the number of generators of this model is sufficiently large and diverse. In this paper we show how the maximal *c*-consensus meets problem can be used to study the databases that generate an integrally private solution. We also introduce a definition of integral privacy based on minimal sets in terms of this maximal *c*-consensus meets problem.

Keywords: Integral privacy · Maximal *c*-consensus meets · Clustering · Parameter selection

1 Introduction

The output of any function computed from a database can be sensitive. It can contain traces of the data used. A simple example is the computation of the mean of a set of salaries. The presence in the database of a person with a high salary may affect the mean salary so significantly that this person presence can be singled out. When the data is sensitive, this type of disclosure can be problematic. This situation applies if we consider an intruder accessing the mean salary of patients in the psychiatric department of a hospital of a small town.

Data-driven models inherit the same problems. Membership attacks [6] are to infer that a particular record has been used to train a machine learning or a statistical model.

Data privacy [7] is to provide techniques and methodologies to ensure that disclosure does not take place. Privacy models are computational definitions of privacy.

Differential privacy [1] is one of the privacy models that focus on this type of attacks. Informally, we say that the output of a function satisfies differential privacy when this output value does not change significantly when we add or remove a record

J. Domingo-Ferrer and K. Muralidhar (Eds.): PSD 2020, LNCS 12276, pp. 62–73, 2020.
https://doi.org/10.1007/978-3-030-57521-2_5

from a database. This means in our case, that the presence or absence of a particular individual in the database cannot be singled out. That is, the presence or absence of the person with a high salary does not change much the *mean* salary. To make differential privacy possible, outputs are randomized so what needs to be similar is the probability distribution on the possible outcomes.

We introduced [8] integral privacy as an alternative privacy model that focuses on the privacy of the generators of the output of a function. Informally, we require that the set of databases that can produce this output is sufficiently large and diverse. We have developed solutions for this privacy model to compute statistics as the mean and deviations [3], decision trees [5] and linear regressions [4]. Our results show that integral privacy can provide in some scenarios solutions with good quality (i.e., good utility) and that solutions may be better with respect to utility than those of differential privacy.

A solution from an integral privacy point of view is one that can be produced by different databases. The more databases that can generate the solution the better, and the more diverse these databases are, the better. Integral privacy formalises this idea.

Our original definition requires databases to be different. In this paper we propose a formalization based on minimal sets. It permits to strengthen our privacy requirements on the generators of the model. This new definition is proposed in Definition 4.

1.1 Model Selection and Integral Privacy

Our definitions of integral privacy are proposed as alternatives to differential privacy. Our goal is to select machine and statistical learning modes that are good from a privacy point of view. Machine and statistical learning is, in short, a model selection problem from a set of candidate ones.

When we need to select a model from candidate solutions (a data-driven model) and we want the model to be integrally private and optimal with respect to other parameters (e.g., accuracy, fairness, bias-free) we need to navigate among sets of records. This is so because each candidate solution needs to be generated by at least a database, and integrally private solutions need to be generated by several alternative databases.

For the sake of explainability we consider that it is appropriate to provide tools to actually explore and navigate on these sets of databases. While from a formal point of view, this is not required, and we can just define a methodology to produce integrally private solutions, as we proceeded in [5], it is convenient to develop techniques to understand the space of databases and, in particular, the databases that generate an integrally private solution. In this paper we study how the maximal c-consensus meets [9] of a set of records can be used for this purpose. The new definition of integral privacy is based on the solution of maximal c-consensus meets. Solutions of this problem represent key records for a given model.

1.2 Structure of the Paper

The structure of the paper is as follows. In Sect. 2 we review integral privacy and in Sect. 3 the maximal c-consensus meets. Then, in Sect. 4, we introduce a new definition for integral privacy based on maximal c-consensus meets. This is the definition on minimal sets. In Sect. 5 we prove some results related to the maximal c-consensus meets

and the related definition for integral privacy. These results explain integrally private solutions in terms of maximal c-consensus meet solutions. Section 6 describes how we compute the solutions. The paper finishes with some conclusions and directions for future research.

2 On Privacy Models: Integral Privacy

In this section we review two privacy models that focus on the privacy of individual records in a database when this database is used to compute a function. That is, we consider a data set X and we apply a function, a query, or an algorithm A to this database and generate an output y. So, $y = A(X)$. Then, we are interested in avoiding inferences on individual records x in X from the information that we have on y.

Differential privacy [1] is one of the privacy models for this type of scenario. Its focus is on the presence or an absence of a record in the database, and that this presence or absence cannot be inferred from y. The definition is based on a comparison of two probability distributions on the space of outputs of A. We assume that different executions of algorithm A may lead to different outcomes, and that the distributions obtained from two databases that differ in only one record, say x, are similar enough. When the distributions are similar enough we cannot infer from the distributions that this particular record x was used.

Definition 1. *A randomized algorithm A is said to be* ε*-differentially private, if for all neighbouring data sets X and* X'*, and for all events* $E \subset Range(A)$*,*

$$\frac{Pr[A(X) \in E]}{Pr[A(X') \in E]} \leq e^{\varepsilon}.$$

In this definition we have that X and X' are neighboring data sets when they differ in one record. We represent that X and X' are neighboring data sets with $d(X,X') = 1$.

Integral privacy [8] is similar to differential privacy in that it focuses on the consequences of knowing the output of a function. I.e., on the inferences that can be obtained from this output. The original definition considered not a single database and a single output but two databases and their corresponding outputs. Then, inferences are not only on the databases but also on the modifications (transitions) that have been applied to one database to transform it into another one. Here we focus on a single database.

The cornerstone of the definition of integral privacy is the concept of generator of an output. That is, the set of databases that can generate the output. We formalize this concept as follows.

Let P be the population in a given domain $\mathscr{D}$. Let A be an algorithm that given a data set $S \subseteq P$ computes an output $A(S)$ that belongs to another domain $\mathscr{G}$. Then for any G in $\mathscr{G}$ and some previous knowledge S^* with $S^* \subset P$ on the generators, the set of possible generators of G is the set defined by $Gen(G,S^*) = \{S' | S^* \subseteq S' \subseteq P, A(S') = G\}$. We use $Gen^*(G,S^*) = \{S' \setminus S^* | S^* \subseteq S' \subseteq P, A(S') = G\}$. When no information is known on S^*, we use $S^* = \emptyset$. Note that previous knowledge is assumed to be a set of records present in the database used to generate G.

Then, integral privacy is about having a large and diverse set of generators. This is to avoid inferences on records in the set and, in particular, avoid membership attacks.

Naturally, it is not always possible to achieve integral privacy because if the only way to generate G from S^* is to have $S^* \cup \{x\}$, previous knowldge S^* implies that $Gen^*(G,S^*) = \{\{x\}\}$. This is the case if we know all patients in the psychiatric department except the rich one. We can infer from G (the mean) that the rich is also in the set.

Definition 2. *Let P represent the data of a population, A be an algorithm to compute functions from databases $S \subseteq P$ into $\mathscr{G}$. Let $G \in \mathscr{G}$, let $S \subseteq P$ be some background knowledge S^* on the data set used to compute G, let $Gen(G,S^*)$ represent the possible databases that can generate G and are consistent with the background knowledge S^*. Then, i-integral privacy is satisfied when the set $Gen(G,S^*)$ is* large *and*

$$\cap_{g \in Gen^*(G,S^*)} g = \emptyset.$$

The intersection is to avoid that all generators share a record. This would imply that there is a record that is necessary to construct G. Following [8] we can distinguish two definitions of *large* in the previous definition.

One follows k-anonymity and requires $Gen(G,S^*)$ to have at least k elements. This means that there are at least k different databases that can be used to build G.

The second definition considers minimal sets in $Gen(G,S^*)$. Let us consider that there are 10 databases that generate a model G. Then, 5 of them share the record r and the other 5 share a record r'. Then, the model G would satisfy at least k-anonymous integral privacy for $k = 2$. In this paper we formalize this second approach in Definition 4.

An important concept in privacy is plausible deniability. We can define it for integral privacy as follows.

Definition 3. *Let G, A, S^*, P, $Gen(G,S^*)$ and $Gen^*(G,S^*)$ as in Definition 2. Integral privacy satisfies plausible deniability if for any record r in P such that $r \notin S^*$ there is a set $\sigma \in Gen^*(G,S^*)$ such that $r \notin \sigma$.*

Naturally, integral privacy satisfies by definition plausible deniability for all records not in S^*. This is so because the intersection of data sets in $Gen^*(G,S^*)$ is the empty set.

Differential privacy and integral privacy have fundamental differences. They are due to the fact that the former requires a *smooth* function, as the addition of any record does not change much the function (i.e., $A(D) \sim A(D \oplus x)$ where $D \oplus x$ means to add the record x to D). In contrast, integral privacy does not require *smoothness* as we do not focus on neighbourhoods. We require that the output of the function for any database results always in what we call *recurrent* models. If $f^{-1}(G)$ is the set of all (real) databases that can generate the output G, we require $A^{-1}(G)$ to be a large set for G. Consider the following example of integrally private function.

Example 1. Let D be a database, let A be an algorithm that is 1 if the number of records in D is even, and 0 if the number of records in D is odd. That is, $f(D) = 1$ if and only if $|D|$ is even.

If this function is applied to an arbitrary subset of the population in Sweden, then the function is integrally private and, therefore, satisfies plausible deniability. The function is not differentially private.

Example 2. Let $P = \{r_1 = 1000, r_2 = 1200, r_3 = 1400, r_4 = 1200, r_5 = 1000, r_6 = 1000, r_7 = 1200, r_8 = 1400, r_9 = 1800, r_{10} = 800\}$ salaries of a population, let $G = 1200$ the mean salary of a database extracted from P. This mean salary will be k-anonymous integral privacy for at least $k = 4$ because the following databases $\{r_1 = 1000, r_2 = 1200, r_3 = 1400\}$, $\{r_4 = 1200\}$, $\{r_5 = 1000, r_8 = 1400\}$, and $\{r_9 = 1800, r_{10} = 800\}$ generate a mean of 1200, and these databases do not share records.

3 Maximal c-consensus Meets

In the previous section we have discussed that the same model can be obtained from different databases. From a privacy perspective, we are interested in these *recurrent* models. Nevertheless, we have also discussed that the recurrence of a model is not enough. When all databases that generate a model share a record, the model is vulnerable to membership attacks.

We have recently introduced [9] maximal c-consensus meets, which can be used to study sets of databases. We show in the next section that this definition permits to define integral privacy in terms of minimal sets.

Given a reference set, the maximal c-consensus meets problem is about finding a set of representatives for a collection of subsets of the reference set. Using notation from lattice theory, we are interested in finding a set of meets that are maximal, in the sense that they have a large number of elements. The problem has similarities (see [9] for details) with other combinatorial optimization problems. In particular, it is related to max-k-intersect, consensus/ensemble clustering, and the minimum set cover problem.

Maximal c-consensus meets is defined in terms of a parameter c, which is the number of representatives we are looking for. It is similar to the number of clusters in clustering algorithms. E.g., k in k-means, c in fuzzy c-means.

3.1 Formalization of the Problem

Let X be a reference set. Let $n = |X|$ be its cardinality, $x_1, \ldots, x_n$ be the elements in X and $\wp(X)$ the set of all subsets of X. The subsets of X define a partially ordered set. Let $A, B \subseteq X$, we use $A \leq B$ when $A \subseteq B$. Therefore, $(\wp(X), \leq)$ is a partially ordered set. I.e., this relationship satisfies reflexivity, antisymmetry and transitivity.

For a partially ordered set $(L, \leq)$, given a subset Y of L we say that $u \in L$ is an upper bound when for all $y \in Y$ we have $y \leq u$. Similarly, $l \in L$ is a lower bound when for all $y \in Y$ we have $l \leq y$. In lattice theory we have the concepts of least upper bound (or join or supremum) and greatest lower bound (or meet or infimum). Then, $(L, \leq)$ is a lattice when each $a, b \in L$ have a join and a meet. We use $\vee$ and $\wedge$ to represent, respectively, the join and the meet as usual. E.g., $a \vee b$ is the join of a and b, and $a \wedge b$ is the meet of a and b.

Given a finite reference set X, the partially ordered set $(\wp(X), \leq)$ is a lattice when the meet is the intersection and the join is the union. This is the lattice we consider in this paper.

Maximal c-consensus meets [9] is defined in terms of a collection S of η subsets of X. Let $S_i \subseteq X$ for $i = 1, \ldots, \eta$, where η is the number of these sets. Then, $S = \{S_1, \ldots, S_\eta\}$. The goal of the problem is to find c parts of the collection whose meets are maximal. Let π_j be a part of S, then, the size of the corresponding meet is $|\cap_{s \in \pi_j} S|$. Let Π be the partition of S with elements π_j for $j = 1, \ldots, c$.

Table 1 gives an example with $X = \{1,2,3,4,5,6,7,8,0\}$ and sets $S_i \subseteq X$ for $i = 1, \ldots, 36$.

When we consider that the total size of the meets of Π is $\sum_{j=1}^{c} |\cap_{s \in \pi_j} S|$ (i.e., the addition of all sizes), we can formalize the maximal c-consensus meets problem as the maximization of the total size of the meets as follows.

$$\begin{aligned} &\text{maximize} \sum_{j=1}^{c} |\cap_{s_i \in \pi_j} S_i| \\ &\text{subject to} \sum_{j=1}^{c} \mu_j(S_i) = 1 \quad \text{for all } i = 1 \ldots \eta \\ &\qquad \mu_j(S_i) \in \{0,1\} \quad \text{for all } i = 1 \ldots \eta \text{ and all } j = 1, \ldots, c \end{aligned} \tag{1}$$

In this formulation μ defines a partition of S. This is so because of the constraints on μ in the problem.

Solutions of the problem above do not require that all meets are large. A few large ones (or all but one large ones) can be enough to lead to a good optimal solution. Because of that, we introduced an alternative definition that we call well-balanced maximal c-consensus meets. In this case we consider the size of the meet with the smallest size. The size of this meet is the one that we want to maximize. The definition follows.

$$\begin{aligned} &\text{maximize} \min_{j=1}^{c} |\cap_{s_i \in \pi_j} S_i| \\ &\text{subject to} \sum_{j=1}^{c} \mu_j(S_i) = 1 \quad \text{for all } i = 1 \ldots \eta \\ &\qquad \mu_j(S_i) \in \{0,1\} \quad \text{for all } i = 1 \ldots \eta \text{ and all } j = 1, \ldots, c \end{aligned} \tag{2}$$

To solve this problem we proposed in [9] the use of a k-means like clustering algorithm and the use of genetic algorithms.

4 Using Maximal c-consensus Meets to Define Integral Privacy

Let P represent the data of a population, A be an algorithm to compute a model (a statistic or a function). Then, different subsets $S \subset P$ will produce models $A(S) \in \mathscr{G}$. Here $\mathscr{G}$ is the space of all possible models.

Let us focus on a particular model $G \in \mathscr{G}$, then $Gen(G, S^*)$ represents all databases that can generate G. From an integral privacy perspective, we are interested in obtaining information on the databases in $Gen(G, S^*)$ that can generate G. The maximal c-consensus meets provide information on this.

Table 1. Set of records corresponding to the problem BC4.

{1, 2, 3, 4, 5, 6, 8, 0}, {1, 2, 3, 4, 5, 6, 8}, {1, 2, 3, 4, 5, 6, 0},
{1, 2, 3, 5, 6, 8, 0}, {1, 2, 4, 5, 6, 8, 0}, {2, 3, 4, 5, 6, 8, 0},
{1, 2, 3, 4, 5, 6}, {1, 2, 3, 5, 6, 8}, {1, 2, 3, 5, 6, 0}, {1, 2, 4, 5, 6, 8},
{1, 2, 4, 5, 6, 0}, {1, 2, 5, 6, 8, 0}, {1, 3, 4, 5, 8, 0},
{2, 3, 4, 5, 6, 8}, {2, 3, 4, 5, 6, 0}, {2, 3, 5, 6, 8, 0},
{2, 4, 5, 6, 8, 0}, {1, 2, 4, 5, 6}, {1, 2, 5, 6, 8}, {1, 2, 5, 6, 0},
{1, 3, 5, 8, 0}, {1, 4, 5, 8, 0}, {2, 3, 4, 5, 6}, {2, 3, 5, 6, 8},
{2, 3, 5, 6, 0}, {2, 4, 5, 6, 8}, {2, 4, 5, 6, 0}, {2, 5, 6, 8, 0},
{3, 4, 5, 8, 0}, {1, 5, 8, 0}, {2, 4, 5, 6}, {2, 5, 6, 8},
{2, 5, 6, 0}, {3, 5, 8, 0}, {4, 5, 8, 0}, {5, 8, 0}

Observe that with respect to maximal c-consensus meets it is irrelevant whether we consider $Gen(G,S^*)$ or $Gen^*(G,S^*)$ as the difference of the corresponding two optimization problems will be the same and the objective functions only differ on a constant.

Observe that Table 1 can be seen from this perspective. Let us consider that the reference set $X = \{1,2,3,4,5,6,7,8,0\}$ represents the individuals of the whole population P and each set in Table 1 represents a database. For the sake of illustration we consider here that when we apply algorithm A to all these databases we obtain the same output.

Then, the maximal c-consensus meets permits us to find clusters of databases that share a large number of records. We will use this perspective to formalize the second definition of integral privacy sketched above. The one that is based on minimal sets in $Gen(G,S^*)$.

Observe that given a set of databases $Gen(G,S^*)$, when we find the optimal partition Π of these databases (in terms of the maximal c-consensus meets) for given a value c, the partition permits us to compute the set of common records $\cap_{s_i \in \pi_j} S_i$ for each $\pi_j \in \Pi$. Let m_j represent this set of common records. Then, from a privacy perspective, a good model G is the one that $m_i \cap m_j = \emptyset$. That is, any pair of meets m_i and m_j share no elements.

This permits to formalize meet-based integral privacy as follows. The definition is based on the parameter c. The larger the c, the larger the privacy. Naturally, if we require a very large c (say 10 or 100) this means that we need to be able to generate the same output with a large number of databases that do not share any record.

Definition 4. *Let P represent the data of a population, A be an algorithm that computes a function from databases $S \subseteq P$ in the set $\mathscr{G}$. Let $G \in \mathscr{G}$, let $S^* \subseteq P$ be some background knowledge on the data set used to compute G, let $Gen(G,S^*)$ represent the possible databases that can generate G and are consistent with the background knowledge S^*, and $Gen^*(G,S^*)$ the same set removing S^* (see definitions above).*

Then, G satisfies c-meets-based integral privacy if there is a solution Π of the maximal c-consensus meets for $Gen^*(G,S^*)$ according to Eq. 2 such that for all $\pi_i \neq \pi_j \in \Pi$ satisfies

$$m_i \cap m_j = \emptyset$$

with $m_i = \cap_{S \in \pi_i} S$ and $m_j = \cap_{S \in \pi_j} S$.

Note that there may be several solutions Π of the optimization problem with the same objective function. We require only that one of them satisfies $m_i \cap m_j = \emptyset$ for all $\pi_i \neq \pi_j \in \Pi$.

This definition implies that a solution G is c-meets based integral privacy if for each $x \neq S^*$ there are at least $c-1$ databases in $Gen^*(G,S^*)$ such that x is not there.

We illustrate this definition with the following example.

Example 3. Note that for $c = 4$, the 4 generators of Example 2 above will satisfy the constraint $m_i \cap m_j = \emptyset$ as we have $m_1 = S_1$, $m_2 = S_2$, $m_3 = S_3$, and $m_4 = S_4$.

5 On the Effects of the Parameter c

Both the maximal c-consensus meets and the definition of integral privacy based on these meets depend on the parameter c. We can study how different parameters c influence the solutions of the optimization problem and the effects on the definition of integral privacy. We first prove results on the objective functions of both optimization problems.

Proposition 1. *For the problem based on addition (Eq. 1), the objective function (OF) is strictly monotonic (increasing) with respect to increasing c. We have* $OF_1 = |\cap_{i=1}^{\eta} S_i|$ *for* $c = 1$*, and* $OF_\eta = \sum_{i=1}^{\eta} |S_i|$ *for* $c = \eta$.

Proof. When $c = 1$, there is a single π_1, and therefore all sets are assigned to it (i.e., $\pi_1 = S$). Therefore, the corresponding meet will be the intersection of all $S_1, \ldots, S_\eta$ and, thus, $OF = |\cap_{i=1}^{\eta} S_i|$.

When $c = \eta$, the optimal assignment is to assign each S_i to a different part. I.e., $\pi_i = \{S_i\}$. In this case, $OF = \sum_{i=1}^{\eta} |S_i|$.

Then, to prove that it is strictly monotonic consider a given c and a given partition $\Pi = \{\pi_1, \ldots, \pi_c\}$ with $c < \eta$ with its corresponding objective function OF_i. Let us consider a part π_i with at least two S_j and S_k assigned to it. As $c < \eta$ such part exists. Then, let define π_i' as π_i without S_j and π_i'' as just S_j (i.e., $\pi_i' = \pi_i \setminus \{S_j\}$ and $\pi_i'' = \{S_j\}$). Finally define a new partition with $c+1$ parts as the previous one replacing π_i by the two new sets π_i' and π_i''. That is, $\Pi' = \{\pi_1, \ldots, \pi_c\} \setminus \{\pi_i\} \cup \{\pi_i', \pi_i''\}$. The cardinality of the meets of π_i' and π_i'' is at least as the same as the cardinality of π_i. Therefore as we add these numbers, the objective function will be larger. □

Proposition 2. *For the problem based on the minimum (Eq. 1), the objective function (OF) is monotonic (increasing) with respect to increasing c. We have* $OF_1 = |\cap_{i=1}^{\eta} S_i|$ *for* $c = 1$*, and* $OF_\eta = \min_{i=1}^{\eta} |S_i|$ *for* $c = \eta$.

Proof. The proof of this proposition is similar to the previous one. We can prove the monotonicity of the objective function using the same sets. Nevertheless, as when we build π_i' and π_i'' from π_i and we include them in the objective function, this objective function just takes the min of the cardinality, the objective function may not strictly increase. E.g., if we have $\pi_i = \{\{1,2,3\},\{1,2,3,4\},\{1,2,3,5\}\}$ and we define $\pi_i'' = \{1,2,3\}$ and $\pi_i' = \{\{1,2,3,4\},\{1,2,3,5\}\}$, the objective function will not increase. □

These two results show that the larger the number of parameters, we have, in general, a larger value of the objective function.

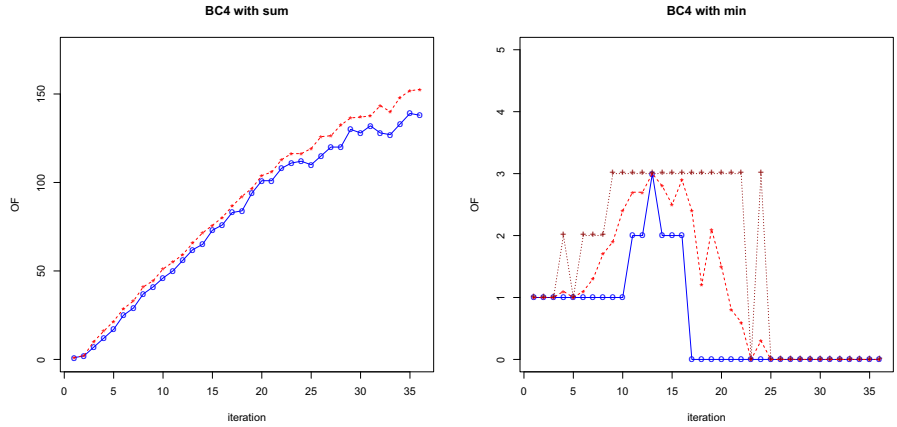

Fig. 1. Values of the objective function for the problem BC4. The figure on left corresponds to the maximal c-consensus meets and the one on the right corresponds to the well-balanced maximal c-consensus meets. Minimum, mean and maximum values of the objective function in each iteration are displayed.

5.1 Open Research Questions

With respect to the definition for integral privacy, it is clear that the larger the c, the more difficult will be to have a model that is compliant with the definition. Nevertheless, an open question is whether a model G that is integrally private for c is also integrally private for any c' such that $c' < c$.

Another open question is whether when a model G with generators $Gen(G,S^*)$ is integrally private for a parameter c, the model is also integrally private when another database is added into the set. That is, if we have two sets of generators $Gen(G,S^*)$ and $Gen(G',S'^*)$ such that $Gen(G,S^*) \subseteq Gen(G',S'^*)$, if integral privacy for $Gen(G,S^*)$ ensures integral privacy for $Gen(G',S'^*)$. We can show with an example that the objective function can decrease when a database is added. It is left as an open problem if this can cause that there is no integrally private solution.

Example 4. Let $A_1 = \{a_1,a_2,a_3\}$, $A_2 = \{a_1,a_2,a_3,a_4\}$, $B_1 = \{b_1,b_2,b_3\}$, $B_2 = \{b_1, b_2, b_3, b_4\}$, $C_1 = \{c_1,c_2,c_3,b_1,b_2\}$, and $C_2 = \{c_1,c_2,c_3,a_1,a_2\}$. An optimal solution for this problem with $c = 3$ is $\pi_1 = \{A_1,A_2\}$, $\pi_2 = \{B_1,B_2\}$, $\pi_3 = \{C_1,C_2\}$. Therefore, $|\pi_1| = |\pi_2| = |\pi_3| = 3$ and the objective function is 3.

If we consider $S' = \{a_1,a_2,b_1,b_2,c_1,c_2\}$ we have that we cannot reach an objective function equal to 3. The assignment $\pi_1 = \{A_1,A_2,C_2\}$, $\pi_2 = \{B_1,B_2,C_1\}$, $\pi_3 = \{S'\}$ results into an objective function equal to 2.

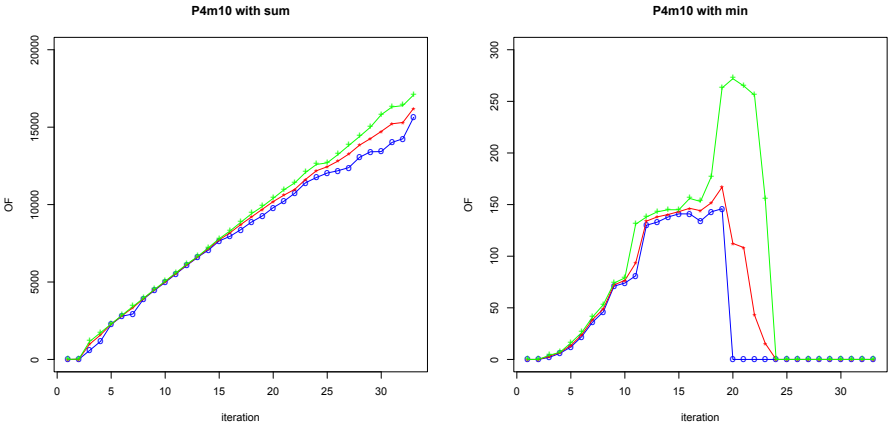

Fig. 2. Values of the objective function for another problem consisting on computing an integrally private *mean* of a file with 1080 records by means of rounding and sampling (following [3]).

6 Solutions Using Genetic Algorithms

In [9] we proposed the use of genetic algorithms to find solutions for the optimization problem described above. We illustrate here the solutions obtained for different values of c. We show the performance of our solution using genetic algorithms when compared with the theoretical results obtained in the previous section. We show that when the objective function uses addition, we obtain results that are consistent with the theoretical ones, but that this is not the case when the objective function uses minimum.

We have considered the solutions obtained for different values of c for the same problems BC1-BC9 considered in [9]. We describe in detail the results of problem $BC4$. This problem consists of the sets of records listed in Table 1. There are 36 sets with at most 9 elements.

We have used 60 iterations with 20 chromosomes each, with a probability of 0.4 for structural mutation and 0.4 for structural crossover. We have used a c that ranges from 1 to η, where η is the number of sets. For $BC4$ we have $\eta = 36$. Each problem is solved 10 times and the minimum, mean, and maximum values of the objective function are recorded. These results for BC4 are displayed in Fig. 1.

It is important to note that while the results above apply for the optimal solutions, solutions found by genetics algorithms are not necessarily optimal. Therefore they do not necessarily satisfy monotonicity. In particular, it is possible that due to structural mutation and structural crossover some of the parts of S are empty, which affects drastically the value of the objective function. Figure 1 gives (left) the results for the first formulation (i.e., Eq. 1) and (right) the results for the second formulation (i.e., Eq. 2).

It can be seen the genetic algorithm is able to obtain results that follow the results in Proposition 1 for the first formulation. That is, the objective function is monotonic with respect to the number of clusters c (except for a few cases). In contrast, our implementation with genetic algorithms does not lead to solutions fully consistent with Proposition 2 for the second formulation. For $c > 15$ the genetic algorithms are not always

able to find the optimal, and for $c > 25$ the genetic algorithms are never able to find the optimal. The second objective function is quite sensitive to parts that are empty and they lead to an objective function with a value equal to zero. Recall that the objective function is the size of the smallest set. In fact, when the number of parts is relatively large there are several parts with no sets associated to them. In addition, the meets of the other parts are still rather small. See e.g., for $c = 23$ we have that the best solution has objective function because there are three parts with zero sets assigned to it, and in addition, even removing these parts the minimum meet is of size only 2. Such *optimal solution* is given in Table 2.

Figure 2 shows another example based on the results explained in [3]. There are 33 sets each consisting of up to 1080 records. These sets are obtained from the computation of an integrally private *mean* of the file by means of rounding and sampling. The details on the file and the computation are found in [3]. We can observe that the results obtained by the algorithms are similar for both problems. In the case of using the objective function with the minimum, the optimal is not achieved for c larger than 20.

Table 2. Optimal solution found for $c = 23$ with the problem BC4.

{1 2 3 5 6 } {} {2 5 6 8 } {0 4 5 8 } {0 3 5 8 }
{} {2 5 6 8 } {0 1 2 4 5 6 } {2 4 5 6 } {0 2 5 6 }
{0 2 3 4 5 6 8 } {0 1 3 5 8 } {0 5 } {1 2 4 5 6 8 }
{5 8 } {} {0 4 5 }{2 5 6 }{0 1 2 5 6 8 }{2 3 5 6 }
{1 5 }{1 2 4 5 6 }{2 3 4 5 6 }

7 Conclusions and Future Work

In this paper we have shown how the maximal c-consensus meets can be used in the context of integral privacy to find the common records of sets of databases that can produce the same solution. We have proven some results related to the monotonicity of the optimal value of the objective function with respect to the number of parts. We have also seen that our approach based on genetic algorithms for solving the optimization problem is not successful for large values of c.

For understanding sets of databases, a smaller c is preferable. How to select a small c with a high objective function is an open problem. We plan to use multiobjetive optimization for this problem.

The maximal c-consensus meets have also been used to formalize a definition for integral privacy. We plan to develop methods to find integrally private models (e.g., decision trees) and statistics (e.g., *means* and *variances*) using the new definition. These solutions need to be evaluated with respect to their utility. Our definition is based on intruder's background knowledge, represented by means of S^*. Further work is needed to analyse what kind of background knowledge can be available.

Acknowledgments. Partial support of the project Swedish Research Council (grant number VR 2016-03346) is acknowledged.

References

1. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_1
2. Rahman, M.A., Rahman, T., Laganière, R., Mohammed, N.: Membership inference attack against differentially private deep learning model. Trans. Data Priv. **11**(1), 61–79 (2018)
3. Senavirathne, N., Torra, V.: Integral privacy compliant statistics computation. In: Pérez-Solà, C., Navarro-Arribas, G., Biryukov, A., Garcia-Alfaro, J. (eds.) DPM/CBT -2019. LNCS, vol. 11737, pp. 22–38. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31500-9_2
4. Senavirathne, N., Torra, V.: Approximating robust linear regression with an integral privacy guarantee. Proc. PST **2018**, 1–10 (2018)
5. Senavirathne, N., Torra, V.: Integrally private model selection for decision trees. Comput. Secur. **83**, 167–181 (2019)
6. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: Proceedings IEEE Symposium on Security and Privacy (2017). arXiv:1610.05820
7. Torra, V.: Data Privacy: Foundations, New Developments and the Big Data Challenge. SBD, vol. 28. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57358-8
8. Torra, V., Navarro-Arribas, G.: Integral privacy. In: Foresti, S., Persiano, G. (eds.) CANS 2016. LNCS, vol. 10052, pp. 661–669. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48965-0_44
9. Torra, V., Senavirathne, N.: Maximal *c*-consensus meets. Inf. Fusion **51**, 58–66 (2019)