Journal of Privacy and Confidentiality (2012) **4**, Number 1, 171–188

Towards Providing Automated Feedback on the Quality of Inferences from Synthetic Datasets

David McClure* and Jerome P. Reiter†

1 Introduction

Many national statistical agencies release data to the public that have been altered to protect the confidentiality of data subjects' identities and sensitive attributes. Unfortunately, for methods of disclosure limitation in practice, it is typically impossible for analysts to gauge how the disclosure limitation has compromised the quality of inferences from the altered data alone. This is particularly problematic when data are intensely redacted to protect confidentiality. Without a sense of data quality, analysts cannot determine if they should trust the analysis; even worse, analysts who blindly trust the results could be led to incorrect conclusions.

Motivated by these problems, Reiter et al. [2009] proposed that agencies create verification servers that provide feedback on the quality of secondary data analyses. The basic idea is as follows. The analyst, who has access only to the altered data, submits a query to the verification server for the results of a statistical model; for example, the coefficients in a regression or the mean of a subpopulation. The server, which has both the confidential and altered data, performs the analysis on both data sources. From the results, the server calculates analysis-specific measures of the fidelity of one to the other. For example, when the query is a regression coefficient, one fidelity measure is the overlap of the 95% confidence intervals for the coefficient when computed with the confidential data and with the altered data [Karr et al., 2006]. The server returns the value of the fidelity measure to the analyst (but not the results of the model from the confidential data). If the analyst feels that the intervals overlap adequately, the altered data have high utility for their analysis. With such feedback, analysts can avoid publishing—in the broad sense—results with poor quality, and be confident about results with good quality [Reiter and Drechsler, 2010].

Reiter et al. [2009] illustrated that fidelity measures provide intruders with information about the confidential data, albeit in a convoluted form, that could be used for disclosure attacks. They suggest general strategies for coarsening fidelity measures to reduce these risks. In this article, we expand on the ideas in Reiter et al. [2009] by examining particular approaches to coarsening fidelity measures. Specifically, we examine approaches based on (i) adding noise to the fidelity measures before release, and (ii) finding interval measures that provide guaranteed levels of safety. We focus on measures specific to multiply-imputed, partially synthetic data [Little, 1993]. This

*Department of Statistical Science, Duke University, Durham, NC mailto:david.r.mcclure@duke.edu

†Department of Statistical Science, Duke University, Durham, NC mailto:jerry@stat.duke.edu

disclosure limitation strategy was only briefly discussed by Reiter et al. [2009], who focused on common disclosure limitation techniques including data swapping, top-coding, and added noise.

The remainder of the article is organized as follows. In Section 2, we review partially synthetic data. We present the parameters of a simulation design and the fidelity measure that we will utilize to empirically demonstrate the different approaches. In Section 3, we describe risks inherent in releasing precise fidelity measures in partially synthetic data and investigate one attempt to reduce those risks: compute the fidelity measure based on data other than that released to the public. In Section 4, we discuss adding random noise to the fidelity measures along the lines of differential privacy output perturbation [Dwork, 2006]. In Section 5, we describe how to release interval fidelity measures and describe their confidentiality properties. Finally, in Section 6 we conclude with some remarks and directions for future research.

2 Partially synthetic data, fidelity measures, and the simulation design

2.1 Partially synthetic data

To illustrate how partially synthetic data might work in practice, we use the setting described by Caiola and Reiter [2010]. Suppose the agency has collected data D on a random sample of 10,000 people. The data comprise each person's race, sex, income, and years of education. Suppose the agency wants to replace race and sex for all people in the sample—or possibly just for a subset, such as all people whose income exceeds \$100,000—to disguise their identities. The agency generates values of race and sex for these people by randomly simulating values from the joint distribution of race and sex, conditional on their education and income values. These distributions are estimated using the collected data and possibly other relevant information. The result is one partially synthetic dataset. The agency repeats this process say ten times, and these ten datasets are released to the public.

To illustrate how a secondary data analyst might utilize these released datasets, suppose that the analyst seeks to fit a regression of income on education and indicator variables for the person's sex and race. The analyst first estimates the regression coefficients and their variances separately in each simulated dataset using standard likelihood-based estimates and standard software. Then, the analyst averages the estimated coefficients and variances across the simulated datasets. These averages are used to form 95% confidence intervals based on the simple formulas developed by Reiter [2003], described below.

Let $D^* = (D_1, \ldots, D_m)$ be the m partially synthetic datasets created by the agency for sharing with the public. Let θ be the secondary analyst's estimand of interest, such as a regression coefficient or population average. For $l = 1, \ldots, m$, let q_l and u_l be respectively the estimate of θ and the estimate of the variance of q_l in synthetic dataset

D_l. Secondary analysts use $\bar{q}_m = \sum_{l=1}^{m} q_l/m$ to estimate θ and $T_m = \bar{u}_m + b_m/m$ to estimate $var(\bar{q}_m)$, where $b_m = \sum_{l=1}^{m}(q_l - \bar{q}_m)^2/(m-1)$ and $\bar{u}_m = \sum_{l=1}^{m} u_l/m$. For large samples, inferences for θ are obtained from the t-distribution, $(\bar{q}_m - \theta) \sim t_{\nu_m}(0, T_m)$, where the degrees of freedom $\nu_m = (m-1)\left[1 + m\bar{u}_m/b_m\right]^2$. Derivations of this inferential method are presented in Reiter [2003] and Reiter and Raghunathan [2007]. Methods for multivariate hypothesis testing are in Reiter [2005b]; methods for handling missing data and partial synthesis simultaneously are found in Reiter [2004] and Kinney and Reiter [2010].

2.2 Fidelity measures

Although synthetic data can preserve associations via modeling, undoubtedly some inferences will deteriorate significantly. These biases may be hard to detect from any meta-data released by the agency describing the synthesis process. Hence, verification servers are arguably essential to the viability of synthetic data products, particularly for high fractions of synthesis.

In this article, we consider queries for scalar estimands and use the interval overlap measure of Karr et al. [2006] as a baseline fidelity measure. For this measure, the server computes the 95% confidence interval for θ from the synthetic data, $Q(D^*) = (L_s, U_s)$, where L_s and U_s are the lower and upper limits of the 95% confidence interval computed using the methods of Reiter [2003]. The server also computes the 95% confidence interval for θ from the confidential data, $Q(D) = (L_d, U_d)$, where L_d and U_d are the lower and upper limits of the 95% confidence interval computed using large-sample normality. Finally, the server computes the intersection between $Q(D^*)$ and $Q(D)$, which we call (L_i, U_i). The fidelity measure is

$$FM(Q(D), Q(D^*)) = \frac{U_i - L_i}{2(U_d - L_d)} + \frac{U_i - L_i}{2(U_s - L_s)}. \quad (1)$$

For the remainder of the article, we abbreviate $FM(Q(D), Q(D^*))$ with FM. When the intervals are nearly identical, corresponding to high utility, $FM \approx 1$. When the intervals do not overlap, corresponding to low utility, $FM = 0$. The second term in (1) is included to differentiate between intervals with $(U_i - L_i)/(U_d - L_d) = 1$ but different lengths. For example, for two synthetic data intervals that fully contain the collected data interval, the measure FM favors the shorter interval.

Many other fidelity measures could be used instead of or in tandem with FM. For example, analysts might be interested in the scaled distance between $\bar{q}_m$ and the point estimate based on the confidential data. We do not consider other measures further here, although the issues we study for FM arise for other measures.

2.3 Simulation design

To illustrate the methods, we use the following simulation design throughout the article. Let the observed data $D = (X, Y)$ be an $n \times 3$ matrix of completely observed data. Here,

X is generated from n independent draws from a standard bivariate normal distribution. For $j = 1, \ldots, n$, each Y_j is randomly drawn from a $N(\beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j}, \sigma^2)$, where $(\beta_0, \beta_1, \beta_2) = (0, 3, -5)$ and $\sigma^2 = 1$. We set $n = 1000$.

We suppose that X is not confidential and can be released to the public without alteration. All values of Y are assumed confidential; thus, we simulate all its values for a public use file. The setting of simulating entire variables is common in applications of partially synthetic data, including the Survey of Income and Program Participation [Abowd et al., 2006] and the Longitudinal Business Database [Kinney and Reiter, 2007]. To generate the synthetic values for each Y_j, we use the posterior predictive distribution that results from the linear regression of Y on X with the standard reference prior distribution, $p(\beta, \sigma^2) \propto 1/\sigma^2$.

When evaluating the risks of providing fidelity measures, we assume that intruders seek to use the fidelity measure to learn about Y. We make the strong assumptions that (i) the intruder knows X for all n records in the database, and (ii) the intruder knows all but one of the units' values of Y. In other words, the only information unknown to the intruder about D is some Y_j. The intruder attacks the verification server to learn this value. This is arguably a worst case scenario; hence, protection strategies deemed safe for this case should be safe for cases with less known information. We also note that for partially synthetic data strategies in which only part of Y is replaced, the intruder can construct queries on subsets of records with only one synthetic Y_j and other unaltered values of Y to create an analogous scenario.

Releasing partially synthetic data entails risks even without the existence of verification servers; for example, intruders could link nonsynthesized values to external files or possibly estimate the masked true values from the released data. For further discussion and examples of risks for releasing synthetic data, see Reiter [2005a], Abowd and Vilhuber [2008], Drechsler and Reiter [2008], and Reiter and Mitra [2009]. Here, we assume that the agency has deemed D^* safe to release, i.e., it is satisfied with the level of disclosure risk in D^*. We also focus exclusively on the information in the fidelity measures about Y_j. We do not account for the joint information about Y_j in both D^* and FM, although we discuss ways to do so in Section 6.

3 Dangers of reporting exact fidelity measures

We now illustrate that reporting infinitely precise versions of FM results in unacceptable disclosure risks. We also examine a method for reducing these risks: computing FM based on data other than D^*.

3.1 Risks in FM when computed with released data

Reporting an infinitely precise FM has many appealing features from an analytical perspective. The reported value gives the exact measure of how much the results for the query on the public data differ from the confidential data, so that the analyst has

the most useful information with which to make a decision about the quality of analysis. It is also easy to compute. However, a verification server that reports the exact FM based on D^* is vulnerable to attack. Let Y_{-j} be the values of Y for all units except unit j, and let D_{-j} be all the values in D except Y_j. For any query Q, the intruder can calculate $Q(D^*)$ from D^*, take a guess at Y_j, say Y_j', and calculate $Q(D_{-j}, Y_j')$. With an exhaustive search of Y_j', the intruder can find the values of $FM(Q(D_{-j}, Y_j'), Q(D^*))$ that correspond to the reported FM.

In this case, the intruder does not need to know how the synthetic data were generated to succeed in the disclosure attack. Further, the intruder can use any query in this attack, even something as seemingly innocuous as the mean of Y. Hence, under our "worst-case" scenario for intruder knowledge, this strategy is too easy to break to be implemented, even with restrictions on the allowable query space.

3.2 Computing FM based on other datasets

The risks in Section 3.1 arise in large part because the intruder knows $Q(D^*)$. This suggests that risks can be reduced if the verification server bases the fidelity measure on different datasets (but D^* is still released to the public). To do so, the agency can generate a new collection of $K*m$ partially synthetic datasets using the same process as for D^*; we call these "ghost" datasets, as they are not to be viewed by the public. To provide maximum utility for analysts, the agency can set K to be as large as computationally feasible. Let $R^* = \{R_1, R_2, \ldots, R_K\}$ comprise K subsets of m distinct datasets. When a query is submitted to the server, for each $R_i \in R^*$, it computes $Q(R_i) = \bar{q}_i \pm t_{\nu_i}\sqrt{\bar{u}_i + b_i/m}$, i.e., the 95% confidence interval based on the m datasets comprising R_i. The server reports a summary of $FM(Q(D), Q(R_1)), \ldots, FM(Q(D), Q(R_K))$, e.g., a list or histogram of the K values, instead of $FM(Q(D), Q(D^*))$. In this way, the attack strategy based on exact matching to FM is no longer possible.

From a utility perspective, the logic behind the ghost datasets approach is frequentist in spirit. The sampled values $FM(Q(D), Q(R_1)), \ldots, FM(Q(D), Q(R_K))$ estimate the distribution of the fidelity measures under the process of generating synthetic data, which reflect how much the particular query could have been affected by the process of generating synthetic data. However, the information loss from using the specific released D^* is not known by the user.

The ghost datasets approach is not immune to attack. The intruder who knows the form of the synthesis model—for example, the agency might release the synthesis code without parameter values as meta-data—can simulate the ghost datasets process to learn about Y_j. Specifically, letting S be the form of the synthesis model, the intruder can compute

$$\begin{aligned} &p(Y_j | FM(Q(D), Q(R_1)), \ldots, FM(Q(D), Q(R_K)), D_{-j}, S) \\ &\quad \propto p(Y_j | D_{-j}, S) \prod_{i=1}^{K} p(FM(Q(D), Q(R_i)) | D_{-j}, Y_j, S) \end{aligned} \tag{2}$$

where $p(Y_j|D_{-j}, S)$ is the intruder's prior distribution. The intruder can use the posterior mode of this density as the best guess at the actual Y_j. Alternatively, the intruder can identify regions with high posterior density to obtain an interval estimate of Y_j.

To illustrate computation of (2), we assume that $p(Y_j|D_{-j}, S)$ is a discrete uniform distribution on a dense grid over a wide range of values of Y_j that includes the true Y_j. We use a discrete distribution to facilitate computation of proper posterior distributions. By making the grid dense and expansive, we can approximate continuous prior distributions as well. For example, to approximate a prior distribution that is a regression of Y_j on X_j with parameters estimated from D_{-j}, we select many values of Y_j at X_j and renormalize their densities.

It is difficult to make general statements about the advantages of any one prior distribution over any other. For example, for values of Y_j far from the regression line, the linear model puts high density on values near the line and low density on values far from the line (near the outlying Y_j), so that it pulls the posterior mode away from the truth compared to using the uniform distribution. But, the linear model provides sharper posterior inferences for values of Y_j that are close to the regression line.

To calculate the posterior distribution based on the discrete uniform prior distribution, we use the following algorithm:

A1. Specify equal prior probabilities on v equally spaced potential values of Y_j between limits a and b.

A2. For each Y_j' given positive weight in step 1, create $D' = (D_{-j}, Y_j')$, and do steps A3 through A5.

A3. Approximate $p(FM(Q(D), Q(R))|D', S)$, where the random variable R represents a draw of m partially synthetic datasets, using a Monte Carlo algorithm as follows. Set $n = 1$.

   (a) Create m synthetic datasets, which we call R_n', by generating them from D' with S. Compute and store $FM(Q(D'), Q(R_n'))$. Set $n = n + 1$.

   (b) Repeat step 3a for N times, where N is as large as computationally feasible; we set $N = 1000$. The collection of N values of $FM(Q(D'), Q(R_n'))$ approximate the sampling distribution of $FM(Q(D'), Q(R))$ given D' and S.

A4. Fit a kernel density estimator to the N sampled values, and estimate the density at each of the reported $FM(Q(D), Q(R_i))$, where $i = 1, \ldots, K$. Call the value of the density $h_i(Y_j')$.

A5. Compute $g(Y_j') = p(Y_j'|D_{-j}, S) * \prod_{i=1}^{K} h_i(Y_j')$.

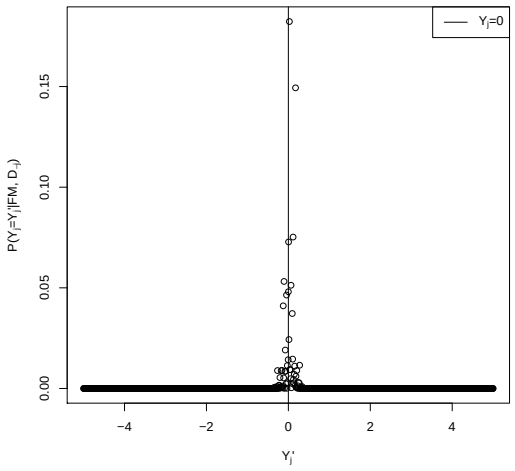

Figure 1: Illustration of concentration of posterior from attack based on A1 – A6 under the ghost datasets paradigm. The query is a regression of Y on $f(X_2)$, where $f(X_2) = (10^6 X_2 - (10^6 X_{2j} - 10^{-6}))^{-1}$ is a transformation that makes X_{2j} massive and all other values of X_2 very small. Here, the target value is $Y_j = 0$, and the reported fidelity measure is based on $K = 20$ ghost datasets and $v = 200$ possible values of Y_j within the limits $(a = -5, b = 5)$.

A6. Once done for all Y_j', we have

$$p(Y_j|FM(Q(D), Q(R_1)), ..., FM(Q(D), Q(R_K)), D_{-j}, S) = \frac{g(Y_j')}{\sum_{Y_j'} g(Y_j')}.$$

In our experience, the success of attacks on ghost datasets varied greatly depending on the queries used. See Figures 1 and 2 for illustrations of the attacks based on steps A1 – A6 using the simulation design of Section 2.3. In general, queries that were amenable to successful attacks, i.e., the posterior mode from the attack is concentrated around the true Y_j, have the properties that (i) $Q(D')$ is sensitive to the value of Y_j', i.e., it changes noticeably with different possible values of Y_j', and (ii) the synthesis process is insensitive to the value of Y_j', i.e., the samples $Q(R_1), ..., Q(R_K)$ are relatively stable. Examples of effective attack queries include the mean of Y based on Y_j and one other observation, and the regression of Y on a version of X transformed so that observation j has extreme leverage in the regression. These are not likely analyses of interest to legitimate data analysts. For queries that work well, the posterior distribution was typically concentrated around a few distinct modes. When information from two or three effective queries were combined, the posterior distribution usually could be narrowed to one mode.

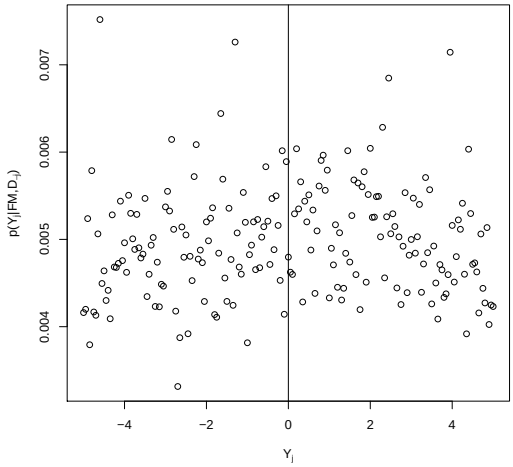

Figure 2: Illustration of dispersion in posterior distribution for Y_j from attack based on A1 – A6 using typical queries under the ghost datasets paradigm. The query is the mean of Y. Here, the target value is $Y_j = 0$, and the reported fidelity measure is based on $K = 20$ ghost datasets and $v = 200$ possible values of Y_j within the limits $(a = -5, b = 5)$.

This attack is computationally expensive. For any query, the intruder must generate $m \times N \times p$ synthetic datasets, where p is the number of points in the support of the intruder's discrete prior distribution. In experiments, we found that intruders need to set N large for attacks to have a good chance of being effective. Applications of synthetic data in large files can take days or even weeks to run a single iteration, so that this computational expense can be a significant disincentive to intruders.

Computational expense also affects agencies when selecting K, so that they may want to set K to be small. Decreasing K also provides less information to the intruder, which could result in lower disclosure risks. However, when K is small, for example $K = 1$, and the distribution of $FM(Q(D), Q(R))$ is not concentrated, there is a high probability that the server could report an FM value that misrepresents the actual quality of the analysis based on D^*.

4 Perturbing outputs by random noise

Since both methods described in Section 3 are vulnerable to attack, we considered adding noise to fidelity measures to increase protection. Instead of reporting $FM(Q(D), Q(D^*))$, the server would report the fidelity measure plus Δ, where Δ followed some distribution with mean zero. Ideally, the amount of noise should be large enough to protect

confidentiality but small enough to offer meaningful reported quality measures. In our experiments, we found that this amount depends on the nature of the query, and we were not able to identify an algorithm that could automatically adapt the amount of noise to the specific query. Additionally, it may be possible to defeat adaptive random noise perturbations using tracker attacks [Dinur and Nissim, 2003].

We therefore examined output perturbation via ϵ-differential privacy [Dwork, 2006], which we define below.

Definition 1 (ϵ-Differential Privacy) *A randomized function $f : D \rightarrow f(D)$ gives ϵ-level differential privacy on data D if*

1. *$\forall$ possible datasets D_1,$D_2 \in D$ that differ on at most one element,*
2. *and $\forall S \subseteq Range(f(D))$,*

$$\Rightarrow \frac{p(f(D_1) \in S)}{p(f(D_2) \in S)} \leq exp(\epsilon). \tag{3}$$

In particular, we considered perturbing the fidelity measures by adding a random draw from the Laplace distribution with parameter $\Delta FM/\epsilon$, where ΔFM is the sensitivity of the fidelity measure. The sensitivity of an output function $f : D \rightarrow \mathbb{R}^d$ is defined as the maximum of $||f(D_1) - f(D_2)||_1$ where D_1 and D_2 differ at most by one observation and $D_1, D_2 \in D$. The parameterization of the Laplace distribution we use is $f(x|\lambda) = \frac{1}{2\lambda} exp(\frac{-|x|}{\lambda})$.

Based on Definition 1, the sensitivity of fidelity measures is difficult to determine analytically. The fidelity measure itself, before subjected to the randomizer, is computed from synthetic data, which itself results from a stochastic process. Thus, for any original data D, the fidelity measure has a distribution that depends heavily on the values in D, the synthesis process, and the query being posed. For the confidence interval overlap measure—and we suspect other fidelity measures—the distribution of the fidelity measure is not amenable to analytical determination; but, it is possible to estimate the distribution empirically for a particular D, synthesis model, and query via simulation. Unfortunately, this does not necessarily enable us to satisfy the first condition of Definition 1, which requires this distribution for all possible datasets (or at least the possible relative difference between distributions resulting from two similar datasets in any part of the domain of D). However, an absolute upper bound on the sensitivity of the fidelity measure for any query and any dataset is one, since fidelity measures are bounded between zero and one.

Therefore, adding $\Delta \sim Lap(1/\epsilon)$ to the fidelity measure will engender at least ϵ-differential privacy. For some queries and types of datasets, it may be that adding Δ engenders differential privacy that is much stronger than ϵ, and that smaller sensitivities could be found. We leave this as a question for future research.

When we add $\Delta \sim Lap(1/\epsilon)$, unless ϵ is large, the size of the noise invalidates the inferential usefulness of the reported fidelity measures. For example, when $\epsilon = 1$, drawing from the corresponding Laplace distribution results $p(|\Delta| > 1) = .37$ and $p(|\Delta| > .5) = .61$

To reduce the sensitivity, it may be possible to apply an approach akin to the method proposed by Smith [2008] for differentially private maximum likelihood estimation. The server can divide D into d disjoint groups, run the query using each subset and the corresponding records in D^* or R^*, and report the average of the d fidelity measures from the subsets. The sensitivity for this approach is at most $1/d$, which results in smaller perturbations. However, for queries based on modest sample sizes, subsetting and averaging fidelity measures can result in values that are quite different from the one calculated using all of D, even before adding noise. Smith [2008] proposed a bias adjustment for maximum likelihood estimation, and a similar adjustment may be useful here.

There is a broader concern with using the Laplace or any other symmetric noise distribution in verification servers. If some user or team of users submits the same query repeatedly, they can average the reported fidelity measures to estimate the original fidelity measure, thus breaking the protection. For popular datasets, we envision verification servers capable of answering hundreds or even thousands of queries per day. It would be a serious drawback to limit the number of queries that analysts can submit. One approach is for the system to decrease ϵ with the number of queries; however, this quickly runs into the problems of uninterpretable output. Another possibility is to give users a privacy budget [McSherry, 2009]. Whether or not this can work for heavily used verification servers in practice is an open question for research. Finally, as suggested by a reviewer of this article, it may be possibile to log queries, so that repeated queries are answered identically every time. This can be challenging to implement in public use contexts, because queries that seem different can be equivalent; for example, the mean age of men and the coefficient from a regression of age on an indicator variable for gender.

5 Releasing intervals as fidelity measures

As an alternative to adding noise to outputs, we explored releasing fidelity measures coarsened to deciles. For example, if $FM = .82$, the server reports the interval $(.8, .9)$. We chose deciles based on our subjective opinion that they provide enough information for data analysts seeking to evaluate quality; for example, the data analyst may not evaluate confidence interval overlaps of 87% and 81% much differently.

Releasing deciles based on FM overcomes many of the shortcomings of the other methods. First, the reported interval is a direct comparison of $Q(D)$ and $Q(D^*)$; it is not perturbed with added noise nor is it based on datasets other than D^* (although deciles could be used in conjunction with ghost datasets). Second, since $FM(Q(D), Q(D^*))$ and thus its containing decile are fixed, data analysts and intruders get the same answer to repeated submissions of any query. This is beneficial for the usefulness and protection

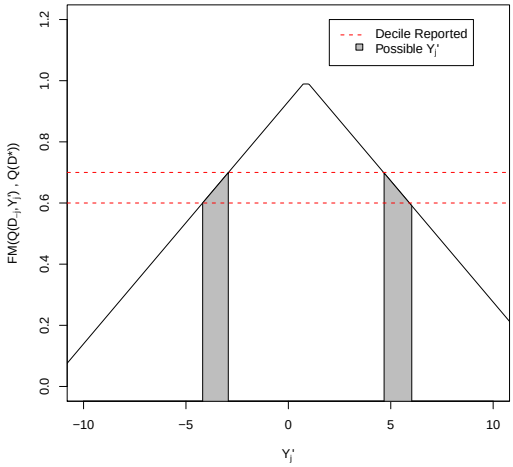

Figure 3: Illustration of attack on fidelity measures reported as deciles, without safety zones. The estimand is the coefficient from a regression of Y on $g(x_2) = (10^6 * x_2 - (10^6 * x_{2j} - 10^{-6}))^{-1}$, with $g(x)$ creating a high leverage point for one particular x_{2j}. The server uses $Y_j = 5$ to calculate $FM = .67$, and reports (.6, .7). The intruder can determine the possible values of Y_j that correspond to an FM in the reported interval (shaded area).

properties of the server. Third, the reported decile leaks no additional information to intruders whose prior beliefs about Y_j have support over values that generate FMs inside the reported decile; that is, letting FMD represent the decile containing FM, for any value of Y_j' that yields a value of $FM \in FMD$, we have

$$p(Y_j'|FMD, D_{-j}) \quad \propto \quad p(FMD|D_{-j}, Y_j')p(Y_j|D_{-j}) \propto p(Y_j'|D_{-j}), \tag{4}$$

since $p(FMD|D_{-j}, Y_j') = 1$ for qualifying values of Y_j.

Despite this promise, reporting deciles alone is not immune to attack. The intruder can plot $FM(Q(D_{-j}, Y_j'), Q(D^*))$ as a function of Y_j'. For some queries, only a small set of values of Y_j' produce an $FM \in FMD$. For example, the queries effective at breaking the ghost datasets approach also are problematic when releasing deciles; see Figure 3. To address this, agencies can establish safety zones for each Y_j; that is, the data owner specifies a range of values, (Y_j^a, Y_j^b), such that it is not considered a disclosure risk if intruders only learn that $Y_j \in (Y_j^a, Y_j^b)$. For example, if continuous $Y_j = 100$, the data holder may deem it acceptable if the intruder only can determine that Y_j is somewhere between $Y_j^a = 92$ and $Y_j^b = 105$. For nominal Y_j, the safety zone is a subset of levels of the categories, so that the intruder can determine only that the actual Y_j must be among the subset. This type of protection for categorical data is akin to PRAM [Gouweleeuw et al., 1998]. Given the safety zones, for any query the server determines the interval to

report as follows:

1. Compute FM and save. Let j = 1.
2. Set $Y_j^s = Y_j^a$. Replace Y_j with Y_j^s to make $D^s = (D_{-j}, Y_j^s)$.
3. Compute $FM(Q(D^s), Q(D^*))$ and save.
4. Set $Y_j^s = Y_j^b$. Replace Y_j with Y_j^s to make new D^s.
5. Calculate $FM(Q(D^s), Q(D^*))$ and save.
6. Return Y_j to its original value. Increment j by one.
7. Repeat steps 2 through 6 for all Y_j, where $j = 1, \ldots, n$.
8. Find the smallest and largest values among all $3n$ saved fidelity measures. Report the lower bound of the decile for the smallest value and the upper bound of the decile for the largest value.

For example, if the lowest fidelity measure is .43 and the highest is .81, the server reports $(.4, .9)$. Figure 4 illustrates this idea graphically for a simple example. This process ensures that the reported interval satisfies the safety zones for each Y_j for any query. For nominal data, the server would replace step 2 through 4 with computation of the fidelity measure for each candidate value of Y_j in the safety zone.

This algorithm can result in reported intervals that are large, even $(0, 1)$ which is useless. However, for most regular queries and reasonable interval sizes, in our simulation we found that the algorithm usually reports intervals of length .10 and rarely goes beyond intervals longer than .20; see Figure 5 for an example. Interval lengths tend to get large for queries that are very sensitive to the value of a particular observation, which are precisely the queries that the server should avoid revealing precise information about.

The algorithm above can and should be improved further, particularly for continuous Y_j. Since the function from Y_j to FM is not monotonic, it is possible that Y_j^a, Y_j, and Y_j^b all produce fidelity measures in the same decile, but many (theoretically all) intermediate values in the safety zone produce fidelity measures outside the decile. If the server reports this decile, the intruder can rule out many intermediate values and is left with several disjoint intervals as candidate regions for Y_j. Thus, the intruder can refine the range of possible Y_j to something much smaller than the safety range, as illustrated in Figure 6. To address this issue, the server could allow Y_j^s to vary over a large number (say P_j) of points interior to the safety zone, in addition to Y_j^a and Y_j^b, as illustrated in Figure 7. The server would report the interval that contains all of the calculated fidelity measures for each examined Y_j^s. The more points included within the safety zone, the less likely substantial regions of intermediate values will fall outside the reported interval. Theoretically, if all the points on the interval (Y_j^a, Y_j^b) were used, the intruder would not be able to eliminate any values in the safety zone as possible values for Y_j.

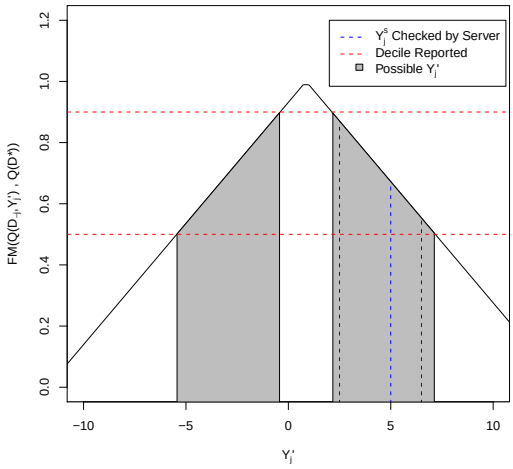

Figure 4: Illustration of safety zone when reporting decile intervals. The safety zone is $(Y_j^a, Y_j^b) = (2.5, 6.5)$. The server releases the interval that contains all FM associated with Y_j^a, Y_j^b, and Y_j. The intruder cannot eliminate any values of Y_j in the gray zone, as desired.

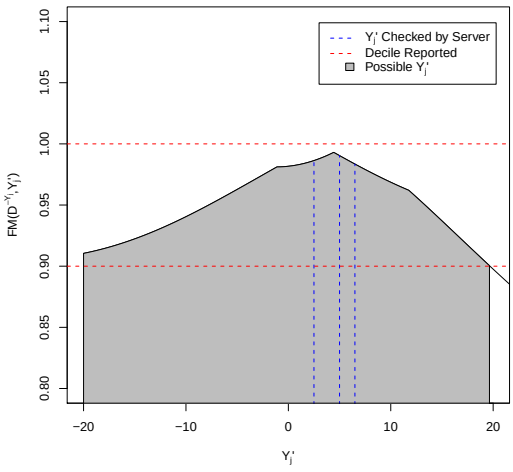

Figure 5: Illustration that normal queries tend to result in FMD of length .10. The query is the coefficient in a regression of Y on X_2. In this case, $Y_j = 5$ and $(Y_j^a, Y_j^b) = (2.5, 6.5)$, so that the decile (.9,1) contains this safety zone.

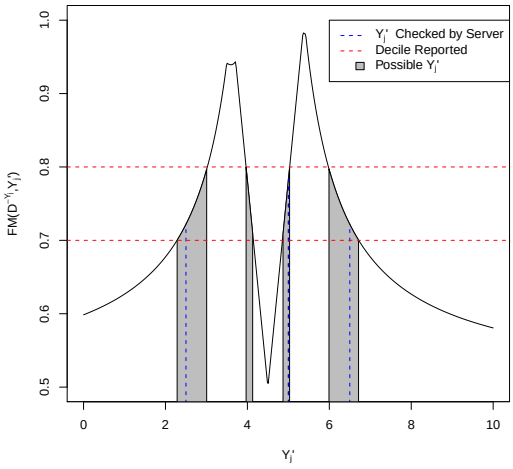

Figure 6: Illustration of potential weakness with safety zone approach. The query is the mean of two elements, $Y_j = 5$ and $Y_{j+1} = 4.5$. The safety zone is $(Y_j^a, Y_j^b) = (2.5, 6.5)$, resulting in FMs of .78, .72, and .72, respectively. The reported decile is (.7,.8), which eliminates many of the intermediate values inside (Y_j^a, Y_j^b). The large dip at 4.5 results because the confidence interval is not computable at 4.5 (estimated standard error equal to zero), and we set $FM(Q(4.5, 4.5), Q(D^*)) = .5$. The dip itself does not give any information on the location of Y_j.

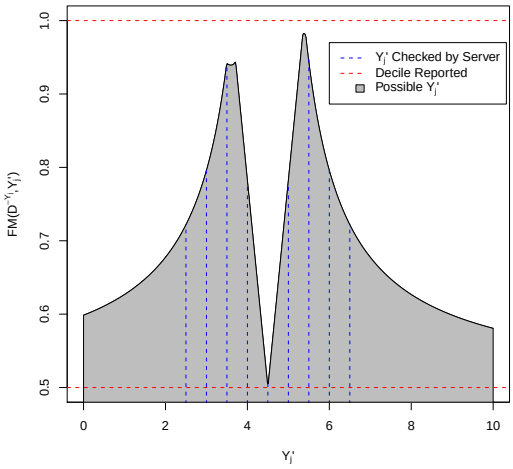

Figure 7: Illustration of safety zone plus checking of interior points. The query is the mean of two elements, $Y_j = 5$ and $Y_{j+1} = 4.5$. We set Y_j^s to include $\{2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5\}$, giving fidelity measures ranging from .5 to .95. The server reports the interval (.5,1). As a result, the intruder cannot eliminate any value in (Y_j^a, Y_j^b) to be Y_j.

Clearly, this would be a demanding approach to implement in practice. Data holders would have to specify the safety zones for each confidential data value. With many synthesized values, the server would have to compute many fidelity measures to come up with safe deciles. This could be prohibitive for queries of complex models. Fortunately, the checks can be easily done with parallel computing, so that the computational burden could be alleviated.

Intruders might be able to determine the safety ranges using the reported deciles, so that agencies must act as if these are known to the public. For continuous Y, it is thus imperative that the safety ranges not be systematically chosen in ways that could be undone. For example, symmetric safety ranges are ineffective because the intruder simply can take the midpoint of the range. It is possible to run the algorithm with ranges that do not contain actual Y_j; simply skip step 1 in the algorithm. It is not clear how much this might impact the quality of the reported deciles.

6 Concluding remarks

When computing posterior probabilities for Y_j, we did not fully use the information in the released synthetic data D^*. Conceptually, this is straightforward; we compute

$$p(Y_j|FMD, D^*, D_{-j}, S) \propto p(FMD|D^*, D_{-j}, Y_j, S)p(D^*|D_{-j}, Y_j, S)p(Y_j|D_{-j}, S). \quad (5)$$

Here, $p(FMD|D^*, D_{-j}, Y_j, S)$ continues to equal one for all values of Y_j in the safety zone. However, $p(D^*|D_{-j}, Y_j, S)$ serves to sharpen the intruder's guess about Y_j before seeing FMD. Hence, D^* itself could make some values in the safety zone more plausible than others, which effectively reduces the protection. In practice, computing $p(D^*|D_{-j}, Y_j, S)$ for complicated synthesis settings can be non-trivial. Further, it is not clear how much information D^* provides about any Y_j, particularly when intruders do not know all of (X, Y_{-j}).

Verification servers arguably are essential for the continued release of public use microdata, especially if high fractions of data are to be altered before release. For verification servers to come to market, they need to share informative measures of data quality that do not leak too much information about the confidential data values. Our investigations with synthetic data and confidence interval overlap measures suggest that releasing carefully constructed deciles have the potential to meet these criteria. However, there is a great deal of research to be done on this topic. How do fidelity measures behave when applied to complex data, where the synthesis models are not so accurate? Also, do our heuristic arguments about the confidentiality guarantees of the decile release algorithm translate to provable privacy guarantees, in the sense that the intruder mathematically cannot learn more about confidential data from the fidelity measure than what the safety zone and D^* tell them? We hope that our work stimulates further research on this important area.

Acknowledgments

This research was supported by NSF grant SES-0751671.

References

Abowd, J., Stinson, M., and Benedetto, G. (2006). Final report to the Social Security Administration on the SIPP/SSA/IRS Public Use File Project. Technical Report, U.S. Census Bureau Longitudinal Employer-Household Dynamics Program. Available at `http://www.bls.census.gov/sipp/synth\_data.html`.

Abowd, J. and Vilhuber, L. (2008). How protective are synthetic data? In J. Domingo-Ferrer and Y. Saygun, eds., *Privacy in Statistical Databases*. New York: Springer-Verlag. 239–246.

Caiola, G. and Reiter, J. P. (2010). Random forests for generating partially synthetic, categorical data. *Transactions on Data Privacy*, 3:27–42.

Drechsler, J. and Reiter, J. P. (2008). Accounting for intruder uncertainty due to sampling when estimating identification disclosure risks in partially synthetic data. In J. Domingo-Ferrer and Y. Saygin, eds., *Privacy in Statistical Databases*, vol. 5262 of *LNCS*. New York: Springer-Verlag. 227–238.

Dinur, I. and Nissim, K. (2003). Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. Association for Computing Machinery. 202–210.

Dwork, C. (2006). Differential privacy. In *Proceedings of the 33^{rd} International Colloquium on Automata, Languages, and Programming, part II*. Berlin: Springer. 1–12.

Gouweleeuw, J. M., Kooiman, P., Willenborg, L., and de Wolf, P. P. (1998). Post randomisation for statistical disclosure control: Theory and implementation. *Journal of Official Statistics*, 14:463–478.

Karr, A. F., Kohnen, C. N., Oganian, A., Reiter, J. P., and Sanil, A. P. (2006). A framework for evaluating the utility of data altered to protect confidentiality. *The American Statistician*, 60:224–232.

Kinney, S. K. and Reiter, J. P. (2007). Making public use, synthetic files of the Longitudinal Business Database. In *Proceedings of the Joint Statistical Meetings*. Alexandria, VA: American Statistical Association.

Kinney, S. K. and Reiter, J. P. (2010). Tests of multivariate hypotheses when using multiple imputation for missing data and partial synthesis. *Journal of Official Statistics*, 26:301–315.

Little, R. J. A. (1993). Statistical analysis of masked data. *Journal of Official Statistics*, 9:407–426.

McSherry, F. (2009). Privacy integrated queries. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD)*.

Reiter, J. P. (2003). Inference for partially synthetic, public use microdata sets. *Survey Methodology*, 29:181–189.

Reiter, J. P. (2004). Simultaneous use of multiple imputation for missing data and disclosure limitation. *Survey Methodology*, 30:235–242.

Reiter, J. P. (2005a). Releasing multiply-imputed, synthetic public use microdata: An illustration and empirical study. *Journal of the Royal Statistical Society, Series A*, 168:185–205.

Reiter, J. P. (2005b). Significance tests for multi-component estimands from multiply-imputed, synthetic microdata. *Journal of Statistical Planning and Inference*, 131:365–377.

Reiter, J. P. and Drechsler, J. (2010). Releasing multiply-imputed, synthetic data generated in two stages to protect confidentiality. *Statistica Sinica*, 20:405–422.

Reiter, J. P. and Mitra, R. (2009). Estimating risks of identification disclosure in partially synthetic data. *Journal of Privacy and Confidentiality*, 1:99–110.

Reiter, J. P., Oganian, A., and Karr, A. F. (2009). Verification servers: Enabling analysts to assess the quality of inferences from public use data. *Computational Statistics and Data Analysis*, 53:1475–1482.

Reiter, J. P. and Raghunathan, T. E. (2007). The multiple adaptations of multiple imputation. *Journal of the American Statistical Association*, 102:1462–1471.

Smith, A. (2008). Efficient, differentially private point estimators. *CoRR*, abs/0809.4794.