Confidentiality Approaches for Real-time Systems Generating Aggregated Results

Jianzhu Li[1], Tom Krenzke[1]
[1]Westat,1600 Research Blvd, Rockville, MD 20850

Abstract

In the past few years many government agencies and statistical institutions have endeavored to release their statistical data through online real-time systems – a high-technology product which is designed to convey information in a timely and flexible manner. In an online real-time system, the data users submit their queries in required format and expect to receive tailored statistical results such as tables with aggregated statistics in a few seconds. Such an environment can accommodate the needs from people with only basic statistical knowledge to sophisticated researchers or statisticians. While a real-time system provides the convenience and flexibility of conducting statistical analyses, it faces the same challenge of protecting data confidentiality as traditional data dissemination approaches – the data products have to be screened and/or treated to ensure a low risk of disclosing an individual's data and/or identity before dissemination. In this paper we review a few approaches to maintain confidentiality and propose an extended approach that is feasible for reducing the disclosure risk in tabular data generated in real-time systems. The performance of the extended approaches is evaluated through the use of risk and utility measures.

Key Words: dynamic, statistical disclosure control, disclosure risk, data utility

1. Introduction

In the past few years there has been growing interest in releasing statistical data, especially aggregated table results, through online real-time system. In such a system, the data users submit their table queries in a required format and expect to receive tailored statistical results quickly, say in a few seconds. Such an environment can accommodate the needs of utilizing restricted data since microdata are not published. However, disclosure risk exists when producing and releasing tabular data. The outputs have to be screened or treated before dissemination. In general, table results based on very small sample sizes have high disclosure risk. Such tables can be restricted or suppressed by simply setting up a threshold rule. But more attention should be given to the attack of forming slivers by table differencing. Slivers are defined as the small differences between two subgroups (or table universes).

One conservative assumption used in differential privacy (Dwork, 2006) is that two subgroups differ by only one case. Assume there is a subgroup A of size *n* and a subgroup B, which is a subset of A, of size *n*-1. In other words, there is only one case which is in subgroup A but not subgroup B while all the other cases are exactly the same in both subgroups. As shown in Figure 1, the same table *X1*X2* (where *X1* and *X2* are variables used to form the cross-tabulation) is produced in subgroup A and B, respectively.

Subgroup A

X1	X2		Total
	1	2	
1	35	45	80
2	55	65	120
Total	90	110	200

Subgroup B

X1	X2		Total
	1	2	
1	34	45	80
2	55	65	120
Total	89	110	199

Figure 1. Table X1*X2 constructed in two subgroups A and B that differ by one case

Typically a simple threshold rule is applied that either denies the table from being shown, or suppresses estimates from being displayed in the cells of the table. Neither of the two tables contains very small cells and would likely satisfy the threshold rule (they do not seem to incur disclosure risk). However, if an intruder takes the difference between the two tables, the resulting sliver table (see Figure 2) reveals the characteristic of the case which is in subgroup A but not in subgroup B, i.e., *X1* = 1 and *X2* = 1. When the tables show unweighted cell counts, the sliver case is in the cell with a count of 1. When weighted frequencies are shown in the tables, the sliver case is in the only non-zero cell with a count equal to its sampling weight, while all the other cells are zeros.

X1	X2		Total
	1	2	
1	1	0	1
2	0	0	0
Total	1	0	1

Figure 2. Sliver table *X1*X2*

After one sliver case is discovered, such an attack can continue by specifying other queries in the same two subgroups, but with different table variables (e.g., *X3* by *X4*, *X5* by *X6*, etc.). Linking the tabular results for the sliver can reveal all the characteristics of that case on the microdata file underlying the system. Moreover, attack can continue by linking the characteristics supposedly protected within the system to other external information in the public domain. This type of attack is called record linking.

2. Perturbation Approaches for Real-time Systems

The basic solution to such a challenge is to blend in noise to the table estimates. Figures 3 and 4 show that, when noise is added to the table cells in Figure 1, table differencing may not be able to reveal the characteristics of the sliver case. The tables produced using the original data are referred to as original tables/estimates, while the tables with noise added are referred to as perturbed tables/estimates. The process of adding noise is called perturbation.

Subgroup A

X1	X2		Total
	1	2	
1	36	44	80
2	56	64	120
Total	92	108	200

Subgroup B

X1	X2		Total
	1	2	
1	33	45	78
2	55	65	120
Total	88	110	198

Figure 3. Table X1*X2 constructed in two subgroups A and B that differ by one case, with noise added

X1	X2		Total
	1	2	
1	3	-1	2
2	1	-1	0
Total	4	-2	2

Figure 4. Sliver table X1*X2, with noise added

The amount of noise added should be controlled by SEEDs that ensure consistent output is produced among users and across time. Such SEEDs can be computed within the system based on the permanent random numbers (PRN) which can be assigned to each microdata record when preparing the underlying data files. Usually the SEEDs are calculated using the functions of the sum of the random numbers. Australia Bureau of Statistics (ABS) (Fraser and Wooten, 2005, Marley and Leaver, 2011) proposed a cell perturbation method to protect confidentiality for tabular output against table differencing attacks. This method adds noise to all table cells independently. A cell-level SEED is used to search for the amount of noise added via a perturbation look-up table. Different functions are considered for generating SEEDs from the permanent random numbers attached to the records in a table cell. The goal of the ABS approach is to produce the same amount of perturbation to table cells containing the same data records. This approach cannot work effectively in the scenario that two universes differ only by one case. For the two *X1*X2* tables in Figure 1, it will produce the same estimates in the three pairs of cells with exactly the same records. Differencing the two tables after adding noise will still result in zeroes in all but one cell in the sliver table.

The Census Bureau considered a protection approach, the Drop-*q* Rule, in their Microdata Analysis System (MAS) that is currently under development (Lucero et. al, 2011). The MAS is being designed to allow users to receive certain statistical output such as tables and regressions using the Census Bureau data. The Drop-*q* Rule is also a SEED-based approach. It deselects a few records (*q*) from the table universe before producing the output. The subsampling process is controlled by the SEEDs generated from the permanent random numbers in the microdata. The Drop-*q* Rule handles mainly unweighted statistics.

Krenzke et al. (2013) extended the use of permanent random numbers and SEEDs by ABS and the Census Bureau to create perturbed tables with weighted estimates and developed a dynamic cell subsampling approach. For this approach, a SEED is generated in each cell using not only the sum of PRNs within that cell, but also the sum of PRNs in the whole table and in each relevant marginal. By doing this, the determination of the

SEED in a cell is dynamic to the query specification of universe and table variables. Although two tables in Figure 1 contain exactly the same records in three out of the four internal cells, different SEEDs will be generated and therefore different amounts of noise will be added since subgroup B contains one case fewer than subgroup A. Within a cell a subsample of records is selected and dropped. Before generating output, survey weights are adjusted at the cell level to account for subsampling and then calibrated to the original universe sum of weights. The same adjustments are done for replicate weights, if there are any, for valid variance estimation. This approach addresses the concern of table differencing and linking attacks successfully.

In this paper we further extend the dynamic cell subsampling approach to account for both unweighted and weighted estimates in tables. The extended approach is named as Drop/add-up-to-*q* algorithm, which allows not only dropping but also adding (e.g. duplicating) up to *q* records in a cell, where *q* is positively correlated with cell size. Dropping a case is equivalent to changing its weight to zero, while adding a case is equivalent to doubling its weight. Dropping or adding exactly *q* records creates the same magnitude of noise in the table cells of similar sizes, which sometimes allows an intruder to infer with high confidence the sliver case's characteristic, especially in tables with unweighted counts. For example, if exactly *q* cases are either added to or dropped from each of the cells in the two tables in Figure 1, then in the sliver table in Figure 2, the cell containing the sliver case will have an unweighted count like $1\text{-}2q$, 1, or $1+2q$ (all odd numbers), while all the other cells will have unweighted counts like $-2q$, 0, or $2q$ (all even numbers). Dropping or adding up to *q* records, in lieu of exactly *q* records, creates uncertainty in the intruder's inference about whether or not there is a sliver, and what the sliver's characteristics are. In this new algorithm, two SEEDs are generated using the sums of random numbers. Again, any changes in cell, marginal, or universe specification of a table should be reflected in the two SEEDs. One SEED is used to control how many records to drop or add in a cell, while the other SEED is used to identify which records to drop or add. For weighted estimates, additional steps are taken to adjust and calibrate the sample weights after adding or dropping records within cells. For example, if a case is dropped, a factor is multiplied to inflate the weights of the remaining cases in the cell. Similarly, a factor is multiplied to deflate the weights if a case is added or duplicated. The weight adjustment ensures the unbiasedness of the table estimates using survey weights. As with the dynamic cell subsampling approach, the same adjustments are done for replicate weights, if there are any, for valid variance estimation.

3. Evaluation

An evaluation study was conducted to demonstrate the use of the Drop/add up-to-*q* algorithm and assess its performance. The National Health Interview Survey (NHIS) 2009 Sample Adult file was first subset to adults with ages 33 to 34. This was to obtain subgroup A with 944 records. Subgroup B*i* is simply defined by removing case *i* (i = 1, …, 944) from subgroup A. Two tables were generated in each subgroup: Age(2) * Sex(2) and Sex(2) * Race(5). Table 1 shows the unweighted counts in the two tables in subgroup A.

Slivers were derived by differencing the table estimates from each pair of subgroups A and B*i*. Without any disclosure protection treatment the differences will reveal the characteristics of the record in the slivers. As a comparison, the Drop/add up-to-*q* algorithm was tailored and applied to protect from a table differencing attack. Up to *q* records were duplicated or dropped in a table cell, where *q* is equal to the smallest integer greater than 1% multiplied by cell size. For example, there are 5 possible treatments in a cell with 198 records: add 1 case, add 2 cases, no add or drop, drop 1 case, and drop 2 cases with each having 20% chance.

Table 1: Unweighted counts in Table Age by Sex in Subgroup A

	Male	*Female*
Age = 33	193	271
Age = 34	201	279

Table 2: Unweighted counts in Table Age by Race in Subgroup A

	Hispanic	*Non-Hispanic White*	*Non-Hispanic Black*	*Non-Hispanic Asian*	*Non-Hispanic Others*
Male	129	218	79	31	7
Female	137	241	71	28	3

By comparing the estimates with and without perturbation, evaluations were done in two perspectives: data utility and disclosure risk. The estimates without perturbation are used as the gold standard. For data utility, we assessed the impact of perturbation on the point estimates and the confidence intervals (e.g., compared the tables in Figure 1 and Figure 3). Ideally the impact of perturbation on utility should be minimal. To measure the disclosure risk of table differencing, we computed the likelihood of correctly identifying the cell membership of sliver cases (e.g., compared the tables in Figure 2 and Figure 4). In a perturbed sliver table, the correct zeros ("0") refer to the cells with zero counts and truly contain no sliver case. The correct one ("1") refers to the cell with a non-zero count and truly contain a sliver case. If all the cells in a perturbed sliver table are either correct zeros or correct one, the table is identified as a correct sliver table. The results are summarized from 944 table differencing attempts of A-B*i*.

For tables with unweighted frequencies, the change in the counts is in general less than 1 percent since the algorithm only drops or adds at most 1 percent of the records in a cell (although this percentage can be higher than that in cells of very small sizes). As a result, the table utility was well retained. In terms of disclosure risk, among the cells in all 944 sliver tables for Age*Sex, there are 20 percent correct zeros and 15 percent correct ones. Overall, 4 percent of the 944 sliver tables have all 4 table cells correct after perturbation. Among the 944 sliver tables, about 83 percent of them have at least 2 out of the 4 cells containing noise added by the perturbation algorithm. In the sliver tables for Sex*Race, there are 29 percent correct zeros and 22 percent correct ones. Compared to the Age*Sex tables, the Sex*Race tables have more cells and therefore relatively smaller cell sizes. The Drop/add up-to-*q* algorithm introduces less noise to the Sex*Race cells, which explains why the Sex*Race tables contain high proportions of correct zeros and correct ones. But the likelihood to having a correct sliver table for Sex*Race is also very low, only less than 1%. More than 85 percent of the sliver tables have at least half of the cells changed after the application of the drop-or-add algorithm.

The risk and utility measures for tables with weighted frequencies are slightly different from those for tables with unweighted frequencies. To measure utility for weighted data, we used the confidence interval (CI) overlap measure (Karr et al., 2006) byfirst computing the overlap in the original and perturbed CIs, and then defining the overlap rate as

$$\frac{1}{2}\left(\frac{overlap\ CI\ length}{original\ CI\ length}+\frac{overlap\ CI\ length}{perturbed\ CI\ length}\right).$$

The closer this measure is to 1, the more data utility is retained. Among the Age*Sex tables, more than 99 percent of the cells have their 95% CI overlap rates greater than 0.955. This indicates that the changes in CIs are almost ignorable after perturbation for majority of the table cells. For the Sex*Race tables, about 90 percent of the cells have their CI overlap rates greater than 0.934. Two out of the ten cells in the Sex*Race table have only a handful of cases. Dropping or adding a case in such cells can cause a large impact on variance estimation, especially when using the paired Jackknife approach (dropping or adding cases with zero weights or doubled weights will cause large change in replicate estimates). Among the other eight cells in the Sex*Race tables, the CI overlap rates are very high. The other utility measure is to compute the relative difference in cell estimates, which is defined as the difference between the original estimate and the perturbed estimate, divided by the original estimate in a specific table cell. For both Age*Sex and Sex*Race, the relative differences are less than 1 percent for all tables.

For tables with weighted frequencies, it is almost impossible to observe exact zeroes in the sliver tables. The perturbed weighted estimate in a cell can be different in subgroup A and subgroup Bi even though this cell contains the same records in the two subgroups and the algorithm determines that one case is dropped within the cell. The noise introduced by perturbation is sensitive to the sample weight of the case being dropped or added. In the sliver tables from differencing weighted estimates, correct zeros refer to the cells with their values close to zero. We used the criterion of the absolute value being less than 2000, which is the 10th percentile of the survey weights in the sample. Correct ones refer to the sliver cells for which the relative difference between the original and the perturbed estimates is less than 10%. The percentage of correct zeros and correct ones are presented in Table 2 for the Age*Sex and Sex*Race tables. Based the above criteria, it is very likely to obtain the correct sliver tables with weighted estimates after applying the Drop/add up-to-q algorithm.

Table 3: Risk measures for tables with weighted estimates

	Correct zeros	*Correct ones*	*Correct tables*
*Age*Sex*	24%	10%	<1%
*Sex*Race*	37%	11%	0%

Another risk measure was developed based on possible attacking scenarios. Assume intruders know the sliver is of size 1 and know the sample weight of the sliver case. Suppose the intruders use two strategies (based on our strong assumptions) to find out the sliver case from table differencing.

- Strategy 1: make a guess that the sliver case is in the cell with the largest positive weight.

- Strategy 2: make a guess that the sliver case is in the cell with the estimate closest to known sample weight.

The success rates for locating sliver cases using Strategy 1 are 49% for Age*Sex and 46% for Sex*Race. The success rates for Strategy 2 are 43% for Age*Sex and 33% for Sex*Race. Although these rates look a bit high, it should be noted that the success rates will deteriorate quickly when linking multiple sliver tables correctly. For example, knowing 6 characteristics of a sliver case requires linking at least 3 two-way tables successfully. Under the two strategies above, the success rate of knowing 6 characteristics drops down to 6.4% if the chance of obtaining one sliver table successfully is 40%.

4. Summary

The Drop/add up-to-*q* algorithm has some advantages over its alternatives when being used in a real-time analytic system for the purpose of disclosure protection. Compared to the traditional cell suppression method, it preserves more useful information in the data. Compared to the controller tabular adjustment and random rounding approaches, it protects against the disclosure risk resulted from not only small table cells, but also table differencing. Compared to the dynamic cell subsampling approach, it can handle not only tables with weighted counts, but also tables with unweighted estimates. It retains the data utility better due to dropping or adding in lieu of only dropping, and dropping/adding possibly fewer than *q* cases in lieu of exactly *q* cases. The disclosure risk is successfully controlled at a low level by introducing more uncertainties to table results. Compared to the approach used by ABS and the Census Bureaus' Drop-*q* method, it provides better protection against table differencing risk since the SEED is not solely cell-based, but also accounts for different universe or marginal specifications.

In general, the Drop/add up-to-*q* algorithm works well to reduce disclosure risk while retaining data utility in a real-time system. It is easy to implement and allows the output to be generated quickly. The use of the SEED guarantees that consistent results are generated for the same queries across time and users. Attention is needed when building this algorithm into a real-time analytic system. It should be tailored based on the nature of the microdata underlying the system to achieve the balance between risk reduction and data quality retention. Also, this algorithm should be used together with other confidentiality approaches in a query system to provide full protection against confidentiality disclosure. Such approaches include requiring the sample sizes in cells, marginals, or table universe to exceed a pre-specified minimum, and limiting the number of variables that can be used to specify universes and/or tables, etc.

References

Dwork, C. (2006). *Differential privacy*. In Bugliesi, M., Preneel, B., Sassone, V. and Wegener, I., eds., ICALP 2006, 1-12. New York: Springer.

Fraser and Wooten. (2005, November). *A proposed method for confidentialising tabular output to protect against differencing*. Paper presented at the UNECE Work Session on statistical data confidentiality.

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.

Krenzke, T., Gentleman, J., Li, J., and Moriarity, C. (2013). Addressing disclosure concerns and analysis demands in a real-time online analytic system. *Journal of Official Statistics*, 29(1), 99-134.

Lucero, J., Zayatz, L., Singh, L., You, J., DePersio, M., and Freiman, M. (2011). The current stage of the microdata analysis system at the US Census Bureau. *Proceedings of the 58th World Statistics Congress of the International Statistical Institute*. isi2011.congressplanner.eu/pdfs/650103.pdf.

Marley, J. K. and Leaver, V. L. (2011) *A method for confidentialising user-defined tables: statistical properties and a risk-utility analysis*. International Statistical Institution: Proceeding 58th World Statistical Congress, Dublin (Session IPS060). http://2011.isiproceedings.org/papers/450007.pdf.