A Proposed 'Bottom-Up' Differential Privacy Approach for Disclosure Prevention in Data Query Tool

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

Abstract

The Occupational Requirements Survey (ORS), conducted by the Bureau of Labor Statistics (BLS) under contract to the Social Security Administration (SSA), collects data on the requirements of work at a detailed occupation level for the overall U.S. civilian economy. BLS and SSA are interested in developing a real-time query system to provide summary tables for the users and researchers. Although ORS is not an establishment survey, the survey data are subject to the risk of disclosing the identifiers of participating establishments if they have some almost unique occupations or their employees consist of a dominantly large proportion of an occupation. A "bottom-up" differential privacy approach was proposed to reduce the risk associated with the published tables from the ORS query tool. A hypercube would be created by cross-tabulating occupation and all work requirements variables, for which noise will be generated from a differentially-private algorithm, adjusted by average quote weight, and added to the records in the hypercube. The hypercube serves as the input data to the query tool. The tables requested by the users will be created by aggregating corresponding records in the hypercube. The hypercube can also be calibrated to the control totals derived from the original ORS data (or with small amount of noise added) at high aggregation levels to reduce the variance of the aggregated noise. For variance estimation, a formula is provided to accommodate both sampling error and perturbation error.

Key Words: Disclosure risk, real-time system, hypercube, perturbation error

1. Introduction

The Occupational Requirements Survey (ORS) is conducted by the Bureau of Labor Statistics (BLS) under contract to the Social Security Administration (SSA). It collects information on the requirements of work at a detailed occupation level for the U.S. economy including physical and mental requirements, as well as education and trainings. The ORS data contain a national probability sample of establishments and occupations, mainly using the Quarterly Census of Employment and Wages (QCEW) to construct the sampling frame. The sample was selected using a two-stage stratified design, where in the first stage establishments were sampled using a probability proportional to the number of employees in the establishments, and in the second stage, occupations were selected from the sampled establishments.

In this paper, a methodology was proposed for developing a real-time query system with underlying ORS microdata to satisfy the needs of BLS and SSA. In the query tool, users can submit requests of weighted tabulations defined by the Standard Occupational Codes (SOC) of establishments and one or more job requirements. The results of estimated total employments and associated standard errors will be displayed within the table cells shortly after the submission.

BLS is mandated to protect the confidentiality of their survey respondents. One of such laws is the Confidential Information Protection and Statistical Efficiency Act of 2002 (CIPSEA). BLS often used certain threshold rules to determine if an estimate is publishable when they release statistical tables. The threshold rules can be based on weighted or unweighted cell frequencies, as well as precision measures such as coefficient of variations. The goal is to achieve the balance between confidentiality protection and the retention of analytic utility.

The vulnerability in the published ORS estimates mainly comes from the 6-digit occupation code, which may indirectly reveal the identities of establishments. The 2018 SOC system (https://www.bls.gov/soc/2018/soc_2018_user_guide.pdf) contains more than 800 detailed occupations. Information known by the user about a specific job code is either well known, or can be inferred from the Quarterly Census of Employment and Wages (QCEW) and the Occupational Employment Statistics (OES). The information could be comprised of a company being the only organization with those jobs, or the company being very dominating in size. The job requirements variables were not considered indirectly identifying. Instead, they were treated as the sensitive attributes of establishments. Once intruders identify a specific establishment through an occupation code, they will know all of the associated job requirements of that establishment.

The potential risk attacks include creating slivers through either direct or indirect approaches. The table cells of very small sample sizes are subject to high risk, especially for sample uniques. Intruders may isolate a data subject by taking differences between tables or linking table with common variables. They may also match the published results to external information through probabilistic record linkage.

2. Differential Privacy

We proposed a solution of reducing the risk by adding noise to ORS estimates through the differential privacy mechanism before displaying results in the query system. With noise being added, the threshold rules will be relaxed and more ORS estimates become publishable. There are two options for adding noise: (1) using an interactive approach for which perturbation will be applied on the fly within the query tool and perturbation is conditional on query specification; (2) using a bottom-up approach for which perturbation can be applied to the most detailed table during data preparation and the perturbed table will be aggregated to generate all the other queried tables. The bottom-up approach was chosen because it satisfies three important properties of a data query tool as follows, whereas the interactive approach may violate table additivity.

- Cell consistency – Across multiple users, if the same set of records contribute to a table cell, the same results are attained;
- Query consistency – Across multiple users using the same query path (e.g., same specification for universe definition and requested table), the same results are attained;
- Additivity – The sum of results from multiple tables is equal to the results directly arrived at for the aggregated table.

Differential privacy is designed to protect against inferences about a unit whether it is in the dataset or not. After treatment, the data release close to zero information to public about a particular individual or contributes almost nothing to re-identification in terms of the mosaic effect (Pozen, 2005). This approach has been implemented to protect tabular data with whole population in companies such as Microsoft, and in the Census Bureau. There has been active research on this topic for different types of data and releases.

Differential privacy makes the disclosure risk pre-determined and measurable. As defined in Dwork, et al. (2006) and Dwork and Roth (2014), a mechanism M satisfies ε-differential privacy if for all neighboring lists a, a' ∈ **A** differing by one individual and all possible outputs b∈ **B**, we have the following likelihood ratio:

$$P(\mathrm{M}(\mathbf{a}) = \mathbf{b})/P(\mathrm{M}(\mathbf{a}') = \mathbf{b}) \leq e^{\varepsilon}.$$

This means that little can be learnt (up to a degree) by an intruder about the target individual that was dropped when moving from a to a'. In other words, the ratio is bounded and the probability in the denominator cannot be zero. Rinott, et al. (2018) proposed using a Laplace mechanism (McSherry, et al. 2007) and proved that this mechanism M for perturbation is ε -differentially private. Define a loss function: $l_1 = l_1(\mathrm{a,b}) = \sum_k | a_k - b_k |$ where a = $\{a_1,..., a_K\} \in \mathrm{A}$ are the original cell counts in the table and b = $\{b_1,..., b_K\} \in \mathrm{B}$ are the perturbed cell counts. The Laplace mechanism is defined as follows: Given a = $\{a_1,..., a_K\} \in \mathrm{A}$, choose b = $\{b_1,..., b_K\} \in \mathrm{B}$ with probability proportional to $\exp[\frac{\frac{\varepsilon}{2}u(\mathbf{a},\mathbf{b})}{\Delta u}]$ where $u(\mathrm{a, b}) = -l_1(\mathrm{a, b})$, ε is the privacy budget controlled by the agency and the scale Δu is defined as: $\max_{\mathrm{b}\in\mathrm{B}} \max_{\mathrm{a}\sim\mathrm{a'}\in\mathrm{A}} | u(\mathrm{a, b}) - u(\mathrm{a', b}) |$ where **a** and **a'** are neighboring databases that differ by removing one individual.

Statistical agencies are concerned about utility when perturbation is applied. One way to ensure high utility in perturbed cell counts is to put a cap on how far away from the original cell counts that are allowed for the perturbation, which leads to the definition of (ε, δ) - differential privacy where δ is the probability of failing to perturb beyond the cap as follows,

$$P(\mathrm{M}(\mathbf{a}) = \mathbf{b}) \leq e^{\varepsilon} P(\mathrm{M}(\mathbf{a}') = \mathbf{b}) + \delta.$$

There is a tradeoff between the two parameters ε and δ .

In principle, there are two ways of developing (ε, δ) - differential privacy for survey weighted employment:

- Perturbation carried out on unweighted cell employments and then the noise adjusted by a factor related to survey weights and added to original weighted cell employments;

- Perturbation carried out on weighted cell employments.

Rinott, et al. (2018) cite future work on applying differential privacy for weighted cell employment and suggest that in this case Δu would be the maximum survey weight. This leads to low utility as the perturbation mechanism M would become quite uniform as opposed to exponential. Defining Δu as the average survey weight (when there is little variability in the survey weights) yields a more exponential perturbation. It also leads to the case of perturbing the unweighted cell employments and then adjusting the perturbed unweighted cell employment through the overall average survey weight (Shlomo, Krenzke and Li, 2018). For example, if the perturbation led to 'add +3 to the original unweighted cell employment', we add 3 times the average survey weight to the original weighted cell employment.

3. A Bottom-up Approach

The flow chart in Figure 1 shows the basic steps of the bottom up approach. It first generates a hypercube from the microdata, then adds noise to the hypercube and control totals. The perturbed control totals are used to calibrate the perturbed hypercube. At the end the calibrated hypercube will be loaded to the query tool. More details are discussed for each step below.

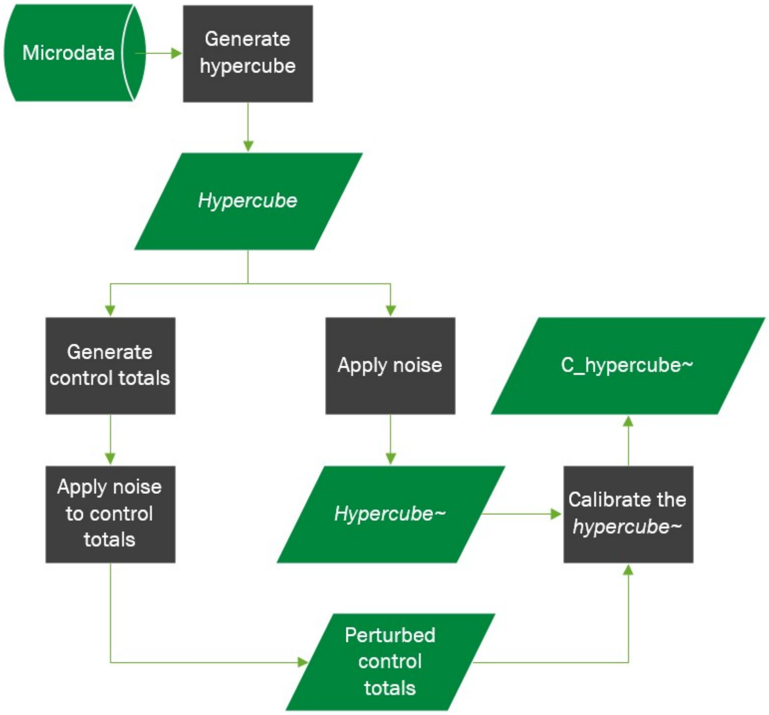

Figure 1. Work flow of the bottom-up approach

3.1 Creation of *Hypercube*

The hypercube is the full cross-tabulation of all the variables that are used to generate the ORS tables, i.e., the occupation code and various job requirements. If too many variables are involved or the variables have too many categories, a large number of cells in the crossed table will contain no sampled cases. In this case, a subsample of those empty cells can be dropped to control the size of the hypercube and save processing time in the next steps. As shown in Table 1, the hypercube cells contained the weighted and unweighted employment counts.

Table 1: Structure of *Hypercube*

Table Cell	*Occupation Code*	*Requirement_1*	*Requirement_2*	…	*Requirement_n*	*Unweighted Employment*	*Weighted Employment*
1	100001	1	1		1	0	0
2	100001	1	2		1	5	100
3	100001	2	1		1	18	500
4	100001	2	2		1	9	300
5	100001	1	2		2	0	0
…	……						

3.2 Perturbing *Hypercube*

Next, noise was generated through a Laplace mechanism. With $\varepsilon = 2$, $\Delta u = 1$ and a cap of ± 7, the perturbation vector and associated probabilities are displayed in Table 2. In expectation, 76.2% of the unweighted counts in the *hypercube* do not change their values, 10.3% of the unweighted counts in the *hypercube* increase or decrease by 1, etc. The unweighted counts can be changed by no more than 7. The δ is then determined by the probability at the cap of ± 7, which in this case is equal to 0.0000006.

Table 2. Perturbation vector and associated probabilities for differential privacy with $\varepsilon = 2$, $\Delta u = 1$ and a cap of ± 7

Perturbation	*Probability of Perturbation*
+/-7	0.0000006
+/-6	0.0000047
+/-5	0.0000346
+/-4	0.0002555
+/-3	0.0018878
+/-2	0.013949
+/-1	0.10307
0	0.76159

Table 3 illustrates how noise was added to cell estimates, or the weighted counts. Noise were generated from the distribution shown in Table 2 and added to the unweighted employment (in terms of number of quotes) in the *hypercube* to obtain the perturbed *hypercube~*. The perturbed weighted employment was calculated as the weighted employment plus the noise multiplied by average quote weight.

Table 3. *Hypercube~*

Cell	*Unweighted Employment*	*Weighted Employment*	*Unweighted Noise*	*Perturbed Unweighted Employment*	*Perturbed Weighted Employment*
1	0	0	+1	1	$\overline{w}$
2	5	100	0	5	100
3	18	500	-2	16	500-2$\overline{w}$
4	9	300	-1	8	300-$\overline{w}$
5	0	0	0	0	0
…	……				

Note: $\overline{w}$ is the average survey weight.

3.3 Calibrating *Hypercube*

The estimates in the ORS query tool will be generated from aggregating the records in the *hypercube~*. As a result, the noise will be aggregated. The more records in the *hypercube~* are aggregated for a query, the more noise is involved in the cell estimates. This is referred to as a "bottom-up" approach in Abowd (2019)[1], where differential privacy is applied to a table at the most detailed level and all aggregations are built form this table. On the contrary, for a "top-down" approach, "differential privacy measurements are taken for tables all levels of details, then large-scale optimization problem is solved to allocate microdata records to solution tables respecting invariants, table consistency, non-negativity, and integer constraints." The "bottom-up" approach is easier to implement and retains table consistency and additivity, while the "top-down" approach, though technically more difficult, has a better control on the differential privacy parameters for tables at all levels.

Calibration can be a remedy for the "bottom-up" approach to reduce the variation in noise for aggregated tables. To reduce the variability of the noise in low dimensional tables, which requires more aggregations, calibration can be done for *hypercube~* to ensure that the noise is controlled at a lower level for select low dimensional table. For example, the *hypercube~* can be post-stratified to the cell estimates of a three dimensional table formed by occupation, education, and prior work experience (other important requirement variables may be used as well). This three dimensional table will first be perturbed through the same algorithm as were done for *hypercube~* with appropriate Laplace parameters. The calibrated hypercube is denoted by *C_hypercube~*. Calibration helps reduce the variation in noise added. To determine the set of control totals, one should consider important high-level tables and the amount of noise added.

3.4 Building Query Tool

When building the query tool, both the original hypercube and the calibrated perturbed hypercube need to be loaded to the system. The calibrated perturbed hypercube is used to calculate the weighted employment in queries. For example, refer to Tables 1 and 3, to calculate the weighted employment in a table cell defined by occupation = 100001 and requirement_1 = 1, simply sum up the perturbed weighted employment of the first two records in Table 1 and Table 3, which meet the conditions. The estimated weighted employment for the cell would be 100+$\overline{w}$. Overall, the records in the *C_hypercube~* can be treated in the same way as any microdata records when calculating the weighted cell employments – but instead of using the weight as you would for microdata, use the perturbed weighted employment as the "weight" when basing the aggregations from the *C_hypercube~*.

For variance estimation, perturbation error needs to be accounted for. The variance of original ORS estimates is estimated using the successive difference replication method. The variance estimator, developed to account for the additional variance due to perturbation, adds a term of squared difference between the original and perturbed estimates to the original variance as follows:

$$\mathrm{var}\left(\tilde{\theta}\right) = \mathrm{var}(\hat{\theta}) + \left(\tilde{\theta} - \hat{\theta}\right)^2,$$

[1] https://www2.census.gov/programs-surveys/decennial/2020/resources/presentations-publications/2019-02-17-abowd-differential-privacy.pdf?

where $\hat{\theta}$ is the estimate from the original hypercube and $\tilde{\theta}$ is the estimate from perturbed hypercube.

A post-perturbation assessment is a critical component of the disclosure limitation process. The assessment evaluates the impact of disclosure protection treatments on reducing risk and retaining data utility. It is important to verify that the perturbed data support valid statistical inference at a similar level to the original data. The assessment results can also be useful for selecting or fine-tuning the parameters of the Laplace mechanism used in the differential privacy technique. To measure the risk, we can check the number of cells that are changed from near zero to non-zero, and vice versa. The more of such changes, the more risk is reduced. To assess the change in utility, we can see how many previously suppressed estimates become publishable, and also check the differences in cell estimates and standard errors. Other common measures include Cramer's V (Agresti, 2002), Hellinger's distance, and confidence interval overlap (Karr et al., 2006).

4. Evaluation

To evaluate the impact of calibration on controlling the noise in mid- to high-dimensional tables, we conducted a small scale simulation study. With a test data of 182 cases, we built a hypercube with 17,280 cells using 10 table variables. Since the sample size is very small compared to the large number of cells in the hypercube, we chose $\varepsilon = 7$ and cap = +/-1 to avoid adding too much noise to the *hypercube*, which basically says, in expectation, noise of +/-1will be added to the cells with 23.5% chances. The variance of the noise is 0.00182.

After noise was generated and applied to hypercube, we calibrated the perturbed estimates to control totals derived from original estimates in all one-way tables (there are 10 of them). We did not add noise to the control totals. The one-way tables only have two to five cells each and would most likely stay unchanged even if a perturbation mechanism were applied with the ε and cap specified above. In practice, it would be good to add noise with a lower value of ε so that these low-dimensional tables are protected.

For this evaluation, the calibration was executed through raking the perturbed estimates to each set of the original control totals iteratively, until convergence was reached. To fully understand the impact of calibration on the variance of noise, we designed the evaluation in the format of simulation. Noise was added to the *hypercube*, independently, for 1,000 times. In the calibration step, we raked the perturbed *hypercube~* to each of the 10 one-way tables and repeated this process for 10 times. If convergence was still not reached after 10 cycles, we did not include this run in the summarized results (convergence may be reached if this process were repeated more than 10 times but we set the limit to 10 to save computational time). Among the 1,000 runs, 224 of them converged successfully. We calculated the variance of noise with and without calibration by different table dimensions (from 1 to 10) in each run, and then took the averages across the 224 runs.

Table 4 shows that, as expected, calibration reduces the variances of noise more effectively in low-dimensional tables than in high-dimensional tables. Since all one-way original table estimates were used for calibration, the noise in the one-way tables generated from the calibrated perturbed hypercube was reduced to almost zero. Among all two-way tables, the variance of noise was reduced by 48.3% on average. The reduction in the variance of noise decreases to 19.3% among all three-way tables and to 5.4% among all four-way tables. The variances of noise were barely changed among all five-way through ten-way tables. It should be noted that the results in Table 4 indicates that variance reduction occurs in expectation, which does not mean that variance reduction will always occur in each of the 224 runs. One of the goals of this evaluation was to see how much calibration would impact the mid-dimensional tables, which we assume may be the most common queried tables.

Table 4. Reduction in Variances of Noise through Calibration by Table Dimensions

Table dimension	*Variance of noise without calibration*	*Variance of noise with calibration*	*Variance reduction*
1	6.592	0.000	100.0%
2	3.255	1.684	48.3%
3	1.340	1.082	19.3%
4	0.522	0.494	5.4%
5	0.200	0.200	0.0%
6	0.077	0.078	0.0%
7	0.030	0.030	0.0%
8	0.012	0.012	0.0%
9	0.005	0.004	0.0%
10	0.002	0.002	0.0%

5. Summary and Conclusions

As discussed in this paper, a non-interactive or bottom-up approach was proposed to build the ORS query tool because this approach can effectively reduce the disclosure risk as well as retaining the additivity property across tables. The bottom-up approach only allows us to set the amount of added noise at an overall level through the parameters of the Laplace algorithm. However, it is difficult to control the noise that is added to individual tables of different dimensions. The interactive differential privacy approach, or adding noise to each individual table based on its specification, may be considered if better data utility is desired, with high tolerance of losing table additivity. When implementing the bottom-up approach, it should be noted that if there are any changes to the microdata (e.g., adding new requirements variables, adding new sample cases), a new hypercube may need to be generated, perturbed, calibrated, and loaded to the query tool. Currently the ORS estimates are mostly weighted counts. More research is needed if means, proportions, or percentiles are of interest in the future.

References

Agresti, A. (2002) *Categorical Data Analysis*, Wiley-Interscience, 2nd ed.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M. (2006) Our Data, Ourselves: Privacy Via Distributed Noise Generation. In: Vaudenay S. (eds) Advances in Cryptology - EUROCRYPT 2006. EUROCRYPT 2006. Lecture Notes in Computer Science, Vol 4004. Springer, Berlin, Heidelberg.

Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, Vol. 9, 211-407.

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.

McSherry, F. and Talwar, K. (2007). Mechanism design via differential privacy. In Foundations of Computer Science (FOCS'07) Proceedings of the 48th Annual IEEE Symposium, IEEE, 94-103.

Pozen, D.E. (2005). The mosaic theory, national security, and the freedom of information act. *The Yale Law Journal*, December 2005, pp. 628–679.

Rinott, Y., O'Keefe, C., Shlomo, N., and Skinner, C. (2018) Confidentiality and differential privacy in the dissemination of frequency tables. To be published in Statistical Sciences.

Shlomo, N., Krenzke, T., and Li, J. (2018). Comparison of Post-tabular Confidentiality Approaches for Survey Weighted Frequency Tables. Presented at Conference on Privacy in Statistical Databases 2018. Valencia, Spain. September 26-28, 2018