Disclosure avoidance for an Online Tax Calculator

The project

The AEI wishes to place on a public webpage a calculator that would allow anonymous users to specify parameters of a US individual income tax law and receive the resulting "distribution table". The calculator would simply apply the specified law to a large database of anonymized tax returns and summarize the resulting tax liability by income class. The tax returns are from the public use file sold by the IRS Statistics of Income Division. A large number of tax parameters would be available, characterizing current law and many plausible reforms.

Although the SOI believes the returns are sufficiently rounded, blurred and aggregated to avoid any disclosure, SOI are concerned that the protections may be insufficient, or that they might be thought to be insufficient. If an intruder were to take advantage of the online calculator to extract information from the PUF file, and then identify (or merely claim to identify) an individual taxpayer that would gravely hurt tax research prospects. Therefore the PUF is currently distributed subject to a non-disclosure agreement and SOI have asked us to protect the underlying micro-data from disclosure via the calculator.

Here I outline how an online tax calculator might lead to disclosure even when cell sizes are large, and how I believe they can be addressed without significantly reducing the utility of the calculator, or inconveniencing users.

Mitigating Factors

I must point out certain mitigating factors that reduce the danger of disclosure, even absent special provisions.

1. Distribution table columns are fixed so only indirectly reflect underlying data items.
2. Limited precison in output means that a single taxpayer would not likely be visible in the difference between two runs.

A differencing attack

In 2008 the distribution table row for taxpayers with AGI greater than $5 million has 2,500 taxpayer records, so at first blush it would seem very resistant to intrusion. The possibility of disclosure arises from the ability of the user to change dollar thresholds. For example the user could set all the tax rates but the top bracket rate to zero, and then try many values for the threshold that begins the top bracket. With sufficient trys, they could identify the highest taxable income in the sample once they found a threshold for the start of the top bracket that just barely generated positive tax. That is, if setting the top (and only) bracket threshold to 200 million dollars, generated no revenue, and 199 million generated some revenue, the taxable income of the highest record must be about 200 million. From there the intruder could identify other characteristics by varying other parameters.

We are going to want to include switches to turn various deductions into credits. By manipulating those controls it would be possible to obtain the amount of itemized deductions for the record already singled out, and if the switch were per deductible item, then the amounts of individual items such as property tax or state income tax would become visible for that record. Although this would not be an actual violation of the law unless the taxpayer were identified by name, it would concern SOI and might be misinterpreted by press or public. A privacy researcher could characterize this process as a disclosure risk, even without identifying a taxpayer. Such publicity is difficult to refute.

All of this could be detected and prevented with some minimal human supervision of run logs, as hundreds of odd looking runs would be required before any information could be extracted from an individual record with techniques of brute force. However users may wish to titrate reforms to a specific revenue target, which might justify repeated runs with similar parameters, so it would not be desirable to limit users to a small number of runs. In any case requests are presumably anonymous, so that it would not be possible to limit runs only for the intruder, rather all users would be adversely affected.

The Census Bureau has developed an effective automated technique for disclosure avoidance in user-specified tables that I believe can be adapted to the case of an online tax calculator. I will quote from *The Microdata Analysis System at the U.S. Census Bureau*, by Freiman, Lucero, Singh and others to describe the Drop *q* rule.

> The drop *q* rule works as follows. A user-defined universe that passes all of the prvious rules has q records removed at random. To do this, the MAS will first draw a random integer value of *q* such that *2<=q<=k* and such that when the universe is modified by omitting *q* records, the number of remaining records is a multiple of 3. Here *k* is some predetermined number, which may depend on the size of the universe. The exact method for determining the maximum possible number *k* of observations to remove is still under consideration. Then, given *q*, the MAS will subsample the universe U(*n*) by removing *q* records at random from the U(*n*) to yield a new subsampled universe U(*n*-*q*).
>
> Within the MAS, all statistical analyses are performed on the subsampled universe U(*n*-*q*) and not on the original universe U(*n*). Each unique universe U(*n*) that is defined on the MAS will be subsampled independently according to the Drop *q* Rule. To prevent an "averaging of results" attack, the MAS will produce only one subsampled universe U(*n*-*q*) for each unique universe U(*n*), with this unique subsample persiting for the lifetime of the system. That is, all users who select a specific universe U(*n*) will have all analyses performed on exactly the same subsampled universe U(*n*-*q*). The MAS accomplishes consistent subsampling of universes by using the same random seed to perform the subsampling every time a given universe comes up. To receive the full disclosure protection offered by the Drop *q* rule, it is necessary that the seed, while constant for a given universe, differs across universes, and this can be implemented by having the seed be a function of the set of units in the universe.
>
> The differencing attacks of most concern require, among other things, that two universes are available that differ in size by 1 or 2. However, under the drop *q* rule described above, all subsampled universes have sizes that are multiples of 3, and no pair of multiples of 3 (including pairs where both numbers are the same) can have a difference of 1 or 2. Hence the Drop *q* rule eliminates the possibility of this sort of disclosure, or even of an apparent disclosure where taking the difference of the resulting tables gives an answer that is palusible (because it has nonnegative numbers in all cells) but is not correct.

How would this apply to an online calculator?

The rule works in conjunction with a requirement that cells have multiple contributing members. As shown above, this is not merely a matter of having a large number of returns with the appropriate AGI for the cell, or contribute a positive amount to the cell, the characteristics of those returns must actually influence the amount of tax owed. For example, everyone could be subject to a poll tax and a single taxpayer subject to additional tax. The intruder could subtract the poll tax from the cell aggregate to be left with the value for the single taxpayer, even though many taxpayers "contributed" to the cell.

I suggest that for every taxpayer we calculate the effect of adding a dollar to income, then keep track of the number of taxpayers in each bracket for which those additions make a difference in the proposed tax liability. This is count of effective cell size. If the count for either of those in any cell is a small positive number (but not zero), that is a potential disclosure that drop-*q* won't prevent. We should also keep track of the number of dropped records which would otherwise contribute to the cell (using this definition). Cells with small numbers of contributors, or very small numbers of dropped potential contributors would have to be suppressed. I don't think that would be a common result, for plausible tax reforms.

Note that a user attempting to difference two runs would find that in addition to the difference he deliberately induced, there would be additional difference provided by the change in the selection of dropped records, and the system would ensure that all of those swapped records were significant contributors to the aggregate AGI.

This doesn't provide absolute protection against all information leakage. It might be possible to identify the state income tax from knowledge of property tax, if (by chance) all the dropped records in a cell were non-itemizers. This strikes me as an acceptable risk, given that those values are already blurred. It would not be possible to bootstrap this

knowledge into information about income items for that record. We could require that some dropped records be itemizers, but there would always be a row in which itemization was rare, and users could always change the itemization rules to make it so.

We would want to ensure that some records were dropped from among the contributors to each cell, without providing any means for the user to exclude non-contributors. So more than *q* records might have to be dropped from some cells, but this is not a problem for our cells are large. The MAS is intended to provide information even when cell sizes are quite small, which is not necessary in our application. Therefore, the concern with keeping the number of records in each cell a multiple of 3 is perhaps an unnecessary complication. Instead, we could drop a larger number of records and rely on the statistical improbability of exactly overlapping samples.

This leaves the intruder with no access to incomes on individual records. Each run is based on a slightly different set of taxpayers so any attempt to isolate a single taxpayer will be frustrated by variation in results from dropping a random set of taxpayers. By testing the effect of a change in income on the results for each taxpayer, we can ensure that these random differences in sample composition are really affecting the results presented to the user.

I do not propose that we expend any effort to ensure that columns aggregate to published totals. This seems to be more of a fetish of agencies rather than a useful feature for data users and is a potential source of bias. However, the average weight of the remaining records in each row should be increased to maintain the sum of weights at the initial value.

Summary

1. We start with the 2009 file - no state id, no extreme values.
2. We drop randomly chosen but effective records from each cell.
3. We track the number of effective records in each cell by checking for differences in tax when income changes.

Daniel Feenberg
feenberg@nber.org
617-863-0343
16 May 2014