Formal Privacy Methods for the 2020 Census

Contact: Gordon Long — glong@mitre.org

April 2020

JSR-19-2F

JASON
The MITRE Corporation
7515 Colshire Drive
McLean, Virginia 22102-7508
(703) 983-6997

Contents

Abstract

In preparation for the 2020 decennial census, the Census Bureau asked JASON to examine the scientific validity of the vulnerability that the Census Bureau discovered in its traditional approach to Disclosure Avoidance, the methods used to protect the confidentiality of respondent data. To address this vulnerability, the Census Bureau will employ differential privacy, a mathematically rigorous formal approach to managing disclosure risk. JASON judges that the analysis of the vulnerability performed by Census is scientifically valid. The use of Differential Privacy in protecting respondent data leads to the need to balance statistical accuracy with privacy loss. JASON discusses this trade-off and provides suggestions for its management.

1 EXECUTIVE SUMMARY

A decennial population census of the United States will officially begin April 1, 2020. Under Title 13 of the US Code, the Bureau of the Census is legally obligated to protect the confidentiality of all establishments and individuals who participate in providing census data. In particular, Census cannot publish any information that could be used to identify a participant.

Over the years, a large amount of personal data have become easily available via online and commercial resources. It has also become much easier to analyze large amounts of data using modern computers and data-science tools. This has made it possible to breach the confidentiality protection promised to respondents of studies and surveys. There have been several notable examples in which records collected under pledges of confidentiality from a survey were linked with public data resulting in the re-identification of the individuals participating in the survey. In an exercise to evaluate the confidentiality protection of the census, the Census Bureau discovered such a vulnerability exists for their data as well.

Using the individual responses from participants (known as microdata), the Census Bureau produces a collection of tables that summarize population counts, age distributions, etc., for various levels of geographic resolution from the whole nation down to census blocks. A variety of approaches have been used by Census in the past to prevent re-identification. In addition to the removal of direct identifiers, Census applies geographic thresholding, top and bottom coding, swapping and other methods of obfuscation to hide identifying characteristics. It was previously thought to be computationally intractable to reconstruct the microdata from the published tabular summaries. But in 2018, applying modern optimization methods along with relatively modest computational resources, Census succeeded in reconstructing, from the published 2010 census data, geographic location (census block), sex, age, and ethnicity for 46% of the US population (142 million people). By linking the reconstructed microdata with information in commercial

databases, Census was then able to match and putatively re-identify 45% of the reconstructed records. Of these putative re-identifications, 38% were confirmed. This corresponds to 17% of the US population in 2010 (a total of over 52 million people). Such a re-identification rate exceeds that obtained in a previous internal Census assessment by four orders of magnitude. Public release of these re-identifications would constitute a substantial abrogation of the Census' Title 13 confidentiality obligations.

In view of these developments, Census has proposed the application of formal privacy methods, in particular, the use of Differential Privacy (DP). DP, introduced in 2006, has as its goal the prevention of learning about the participation of an individual in a survey by adding tailored noise to the result of any query on data associated with that survey. DP provides a set of algorithms used to compute statistical information about a dataset (e.g. counts, histograms, etc.), but infuses those statistics with tailored noise, making it possible to publish information about a survey while limiting the possibility of disclosure of detailed private information about survey participants.

A number of features make DP an attractive approach for protection of confidentiality for the 2020 census and beyond. Notably, privacy loss (in a technical sense) can be rigorously quantified via a privacy-loss parameter. In addition, there are techniques to create synthetic data such that subsequent queries will not cause further confidentiality loss provided such queries do not access the original data. Finally, confidentiality would degrade in a controlled way should it prove necessary to re-access the original data in order to publish further tabulations. Census proposes to use this approach by adding noise to the tabular summaries it traditionally publishes and then using these to reconstruct synthetic census microdata. Both the noised tabular summaries and the synthetic microdata could then be publicly released.

Once the differentially private tabulations and the synthetic data are produced, the use of DP methods offers a mathematically rigorous guarantee that any

further analysis of the released data preserves the original level of confidentiality protection. However, one drawback of such approaches is that the applied noise will degrade the accuracy of various tabulations and statistical analyses of the data, particularly those associated with small populations. Census data are used by a large number of government, academic, business, and other stakeholders. Census is therefore compelled to make an explicit trade-off between the accuracy of its data releases and the privacy of respondents.

Census charged JASON with the following three tasks:

1. Examine the scientific validity of the vulnerability that the Census Bureau discovered in the methods that it has historically used to protect the confidentiality of respondent data when preparing publications;
2. Evaluate whether the Census Bureau has properly assessed the vulnerability as described above;
3. Provide suggestions to represent the trade-offs between privacy-loss and accuracy to explicitly represent user choices.

JASON has not attempted to duplicate the reconstruction of census microdata as it does not have access to that data, nor to data from commercial marketing databases. JASON has, however, confirmed via database simulation that such an attack is possible; JASON concludes that, provided one publishes a sufficient number of tabular summaries, there are multiple approaches using modern optimization algorithms to reconstruct the database from which the summaries originated with high probability. This creates a significant risk of disclosure of census data protected under Title 13.

Census plans to release some data without noise, most importantly, state populations for the apportionment of Congressional representatives. In addition, Public Law 94-171 requires that Census provide the states with small-area data necessary to perform legislative redistricting for both Federal and State electoral

districts. The Census has set up a voluntary program in which state officials define the geographic areas for which they wish to receive census data. While only population data are legally mandated, Census has traditionally also provided other demographic data such as race, ethnicity and voting age populations. For expedience, states have simply asked for these data at the finest geographical resolution (census blocks) and have then used the block populations to infer population counts for larger geographical areas such as legislative districts. The proposed creation of differentially private census tabulations will result in block-level populations that differ from the original census enumeration due to the infused noise. Releases of exact counts (known as invariants) are technically violations of DP in principle and degrade the privacy guarantee, although to what extent in practice remains a research issue. There arises, then, a tension between the obligations under PL 94-171 to release population data for legislative purposes and the requirements of Title 13 to protect confidentiality.

For large populations, for example at the national, state, or even in many cases the county level, using DP does not unduly perturb the accuracy of statistical queries on the data provided the privacy-loss parameter is not set too low (implying the infusion of a large amount of noise). This is important for diverse users of census data (demographers, city planners, businesses, social scientists etc.). But as the size of the population under consideration becomes smaller, the contributions from injected noise will more strongly affect such queries. Note that this is precisely what one wants for confidentiality protection, but is not desirable for computation of statistics for small populations. Thus there is also a tension between the need to protect confidentiality and the aim to provide quality statistical data to stakeholders. While the latter is not legally mandated for Census, it is aligned with the Office of Management and Budget's policy directive to agencies that produce useful governmental statistics, and Census has traditionally been a key supplier of such data through its various published products.

The trade-off between confidentiality and statistical accuracy is reflected in the choice of the DP privacy-loss parameter. A low value increases the level of

injected noise (and thus also confidentiality) but degrades statistical calculations. Another factor that also influences the choice of privacy-loss parameter is the number and geographical resolution of the tables released. For example, if no block-level data were publicly released, a re-identification "attack" of the sort described above presumably would become more difficult, perhaps making it feasible to add less noise and thus publish tables at a higher value of the privacy loss parameter than what would be required if block level tables were published. A re-identification attack, of the sort that originally led to the conclusions that more rigorous and effective confidentiality protections were required, has not been performed on microdata reconstructed from differentially private tabulations. Such an analysis is needed to gauge the level of protection needed.

Depending on the ultimate level of privacy protection that is applied for the 2020 census, some stakeholders may well need access to more accurate data. A benefit of differential privacy is that products can be released at various levels of protection depending on the level of statistical accuracy. The privacy-loss parameter can be viewed as a type of adjustable knob by which higher settings lead to less protection but more accuracy. However, products publicly released with too low a level of protection will again raise the risk of re-identification. One approach is to use technology (e.g. virtual machines, secure computation platforms etc.) to provide protected data enclaves that allow access to census data at lower levels of privacy protection to trusted stakeholders. Inappropriate disclosure of such data could still be legally enjoined via the use of binding non-disclosure agreements such as those currently in Title 13. This idea is similar to the concept of "need to know" used in environments handling classified information.

Finally, it will be necessary to engage and educate the various communities of stakeholders so that they can fully understand the implications (and the need for) DP. These engagements should be two-way conversations so that the Census Bureau can understand the breadth of requirements for census data, and stakeholders can in turn more fully appreciate the need for confidentiality protection in the present era of "big data", and perhaps also be reassured that their statistical

needs can still be met.

1.1 Findings

1.1.1 The re-identification vulnerability

- The Census has demonstrated the re-identification of individuals using the published 2010 census tables.
- Approaches to disclosure avoidance such as swapping and top and bottom coding applied at the level used in the 2010 census are insufficient to prevent re-identification given the ability to perform database reconstruction and the availability of external data.

1.1.2 The use of Differential Privacy

- The proposed use by Census of Differential Privacy to prevent re-identification is promising, but there is as yet no clear picture of how much noise is required to adequately protect census respondents. The appropriate risk assessments have not been performed.
- The Census has not fully identified or prioritized the queries that will be optimized for accuracy under Differential Privacy.
- At some proposed levels of confidentiality protection, and especially for small populations, census block-level data become noisy and lose statistical utility.
- Currently, Differential Privacy implementations do not provide uncertainty estimates for census queries.

1.1.3 Stakeholder response

- Census has not adequately engaged their stakeholder communities regarding the implications of Differential Privacy for confidentiality protection and statistical utility.
- Release of block-level data aggravates the tension between confidentiality protection and data utility.
- Regarding statistical utility, because the use of Differential Privacy is new and state-of-the-art, it is not yet clear to the community of external stakeholders what the overall impact will be.

1.1.4 The pace of introduction of Differential Privacy

- The use of Differential Privacy may bring into conflict two statutory responsibilities of Census, namely reporting of voting district populations and prevention of re-identification.
- The public, and many specialized constituencies, expect from government a measured pace of change, allowing them to adjust to change without excessive dislocation.

1.2 Recommendations

1.2.1 The re-identification vulnerability

- Use substantially equivalent methodologies as employed on the 2010 census data coupled with probabilistic record linkage to assess re-identification risk as a function of the privacy-loss parameter.
- Evaluate the trade-offs between re-identification risk and data utility arising from publishing fewer tables (e.g. none at the block-level) but at larger values of the privacy-loss parameter.

1.2.2 Communication with external stakeholders

- Develop and circulate a list of frequently asked questions for the various stakeholder communities.
- Organize a set of workshops wherein users of census data can work with differentially private 2010 census data at various levels of confidentiality protection. Ensure all user communities are represented.
- Develop a set of 2010 tabulations and microdata at differing values of the privacy-loss parameter and make those available to stakeholders so that they can perform relevant queries to assess utility and also provide input into the query optimization process.
- Develop effective communication for groups of stakeholders regarding the impact of Differential Privacy on their uses for census data.
- Develop and provide to users error estimates for queries on data filtered through Differential Privacy.

1.2.3 Deployment of Differential Privacy for the 2020 census and beyond

- In addition to the use of Differential Privacy, at whatever level of confidentiality protection is ultimately chosen, apply swapping as performed for the 2010 census so that no unexpected weakness of Differential Privacy as applied can result in a 2020 census with less protection than 2010.
- Defer the choice of the privacy-loss parameter and allocation of the detailed privacy budget for the 2020 census until the re-identification risk is assessed and the impact on external users is understood.
- Develop an approach, using real or virtual data enclaves, to facilitate access by trusted users of census data with a larger privacy-loss budget than those released publicly.

- Forgo any public release of block-level data and reallocate that part of the privacy-loss budget to higher geographic levels.
- Amid increasing demands for more granular data and in the face of conflicting statutory requirements, seek clarity on legal obligations for protection of data.

2 INTRODUCTION

2.1 Overview of the Census

The US decennial census, codified in law through the US Constitution has taken place every 10 years since 1790. The 24th such census will take place in 2020. The authority to collect and analyze the information gathered by the Census Bureau originates in Title 13 of the US Code enacted into law in 1954. Title 13 Section 9 of the US code mandates that neither the Secretary of Commerce or any other employee or officer of the Dept. of Commerce may

> "... use the information furnished under the provisions of this title for any purpose other than the statistical purposes for which it is supplied; or make any publication whereby the data furnished by any particular establishment or individual under this title can be identified; or permit anyone other than the sworn officers and employees of the Dept or bureau or agency thereof to examine the individual reports."

Census employees are sworn to uphold the tenets of Title 13 and there are strict penalties including fines and imprisonment should there be any violation. To ensure the mandate of Title 13 is upheld, the Census has traditionally used what are termed Disclosure Avoidance techniques on its publicly released statistical products. The particular approaches used by the Census for Disclosure Avoidance have evolved over the years. A short overview is contained in this report.

Surveys have long been an invaluable tool in determining policy and in the performance of social science and demographic research. In many cases such surveys require respondents to reveal sensitive information under the promise that such information will remain confidential. Traditionally, protection from disclosure was accomplished by anonymizing records. In this way, statistical analyses on issues of public importance could be accomplished while protecting the identity of the respondent. Over time however, the availability of public external data

and the increase in capability of data analytics has made protecting confidential data a challenge. By linking information in one data set with that of another containing some intersecting information (known as a record-linkage attack) it is sometimes possible to connect an anonymous record containing confidential information with a public record and thus identify the respondent. This is called re-identification of previously de-identified data. A number of newsworthy re-identifications have been accomplished in this way. Several approaches have been put forth to make such record linkage attacks harder (see e.g., [32]) but to date none of these have proven to be sufficiently robust to attack.

In 2016, analysts at the Census realized that, even though the Census publishes for the most part tabular summaries of its surveys, enough information could be gleaned from the results to correctly reconstruct a substantial fraction of the detailed survey responses. By linking this information with commercial marketing databases, the names of the respondents could be ascertained, a putative violation of Title 13.

In response, Census has proposed to utilize methods of formal privacy developed and analyzed in the cryptography community; Census proposes to use the methods of Differential Privacy (DP) [8] to secure the 2020 Census. Census requested a JASON study as part of the process of verifying their assessment of disclosure risk as well as assessing the proposed use of formal privacy approaches. Census' charge to JASON was as follows:

- JASON will examine the scientific validity of the vulnerability that the Census Bureau discovered in the methods it has historically used to protect the confidentiality of respondent data when preparing publications.
- Risk assessment: has the Census Bureau properly assessed the vulnerability?
- Implementing formal privacy requires making explicit choices between the accuracy of publications and their associated privacy loss; users always

want more accuracy, but the Census Bureau must also safeguard the respondents' privacy. How do we represent the trade-offs between privacy loss and accuracy to explicitly represent user choices? Are there other conceptual approaches we should try?

2.2 Overview of the Study

JASON was introduced to the relevant issues through a set of presentations listed in Table 2-1. The briefers were experts both internal and external to the Census Bureau in areas such as disclosure avoidance, demography, and applications of census data such as redistricting. These talks were of high quality and were instrumental in educating JASON on these issues. In addition, members of JASON participating in the study were sworn into Title 13 allowing them to be briefed on information protected under this statute and providing JASON with important insights into the details of 2020 Census and particularly the Disclosure Avoidance system based on DP proposed for 2020. Finally, Census provided with JASON with a rich set of reference materials, some protected under Title 13. Details associated with those materials protected under Title 13 are not included in this report.

2.3 Overview of the Report

In Section 3, we provide a brief overview of the census process, the information that Census is mandated to provide and the associated timeline. We also briefly review the methods that were used for Disclosure Avoidance in the past. In Section 4, we review the work that led Census to conclude that the previous approaches to Disclosure Avoidance were inadequate given the increasing availability of large datasets of personal information. In this context, we discuss the seminal work of Dinur and Nissim [5] leading to what is now called the Fundamental Law of Information Recovery. We also describe some experiments asso-

Speaker	Title	Affiliation
Ron Jarmin	Overview of the Dual Mandate and Legal and Historical Background for Disclosure Avoidance	US Census
Victoria Velkoff	Proposed 2020 Census Data Products	US Census
James Whitehorne	Overview of Redistricting Data Products	US Census
John Abowd	The Vulnerability in the 2010 Census Disclosure Avoidance System (DAS)	US Census
Ashwin Machanavajjhala	Interpreting Differential Privacy	Duke University
Dan Kifer	Design Principles of the TopDown Algorithm	Penn State University
Phil Leclerc	Empirical Analysis of Utility-Privacy Trade-offs for the TopDown Algorithm	US Census
William Sexton	Disclosure Avoidance At-Scale and Other Outstanding Issues	US Census
Cynthia Hollingsworth	How 2020 Census Data Products are Prepared	US Census
Rachel Marks	How 2020 Census Data Products Reflect Data User Feedback	US Census
Ken Hodges	How 2020 Census Products will be used by Demographers	Claritas
Justin Levitt	Uses of 2020 Census Redistricting Data	Loyola University
Tommy Wright	Suitability Assessment of Data Treated by DA Methods for Redistricting	US Census
Kamalika Chaudhuri	Formal Privacy and User-Imposed Constraints	UCSD
Salil Vadhan	Formal Privacy and Data Analysis, Including Invariants	Harvard
Dave van Riper	Differential Privacy and the Decennial Census (via VTC)	U. Minnesota
Danah Boyd	Video Teleconference	Microsoft
Jerry Reiter	Video Teleconference	Duke University

Table 2-1: Briefers for JASON Census study.

ciated with the Dinur-Nissim work that underscore the conclusions of that work. In Section 5, we describe briefly the proposed use of DP as a means of protecting sensitive Census data. DP grew out of the work described above by Dinur and Nissim and then extended by Dwork and her collaborators [7]. DP makes possible statistical queries regarding a dataset to be performed while offering a rigorous bound on the amount one learns about a dataset if one record is deleted, added or replaced. Note that this is not, strictly speaking a guarantee of disclosure avoidance but it does provide in a rigorous way the likelihood of a record linkage attack. It does this by adding specially calibrated noise to the result of a specific query made on the dataset. For queries that involve large populations, the addition of noise does not unduly perturb the statistical accuracy of the query. But as a query focuses on smaller and smaller populations the noise will make it increas-

ingly difficult to infer individual characteristics. An attractive feature of DP is that the level of protection is tunable via the setting of a privacy loss parameter. The value set for the privacy loss parameter is meant to be a policy decision.

In Section 6, we discuss the results of some of the early work performed by Census on applying DP to census data. Census proposes to use DP to process the sensitive microdata and create the standard tabular summaries. Noise will then be added to these summaries to make them differentially private. The assessment of the privacy loss budget to be used has not yet been performed. Census will then use the same reconstruction algorithms it applied on the 2010 census data on the noised tables. This will create synthetic microdata that, in principle, should be safe to publish openly. We discuss some early applications of this approach and the nature of the synthetic data it produces. The proposed use of DP will lead to tension between protecting privacy while providing accurate demographic data for activities like redistricting. In Section 7 we propose some approaches for managing this trade-off. Finally in Section 8 we summarize our findings and recommendations.

3 CENSUS PROCESS

In this section we provide a brief overview of the main products that the Census provides as well as the geographic hierarchy that Census has established to collect the relevant respondent data. We also cover the approach the Census has used to process and summarize the required data. Finally, we discuss the evolving need for preservation of the confidentiality of Census data.

3.1 Census Geographical Hierarchy

The Census organizes the US population via a geographic hierarchy shown in Figure 3-1. At the top of this hierarchy are the national boundaries of the United States and Puerto Rico. Within each state, Census further subdivides the population according to county of residence. Counties are then further divided into tracts, block groups, and finally the lowest gradation of Census geography, the Census block. Census also surveys the households in each block and counts for example the number of residents, whether the resident owns or rents etc. Census also collects data for what are known as Group Quarters. Examples of these are dormitories, prisons, etc. The designations in Figure 3-1 of nation, region, state, county, tracts, block groups, and finally census blocks is called the “central spine" of the census geographic hierarchy. Off this spine are also indicated other important state and local divisions. For these, Census provides geographies that can then be used to determine counts in these regions off the spine. These Census geographies inform the placement of Census blocks so that the counts in these areas can be performed from Census block data.

The distribution of population and the number of households in a census tract, block group or block varies greatly across the nation. A map of the population density from 2010 census data is shown in Figure 3-2. As can be seen, the population density varies from thousands of people per square mile as for example in areas like New York City or Los Angeles down to less that ten people per

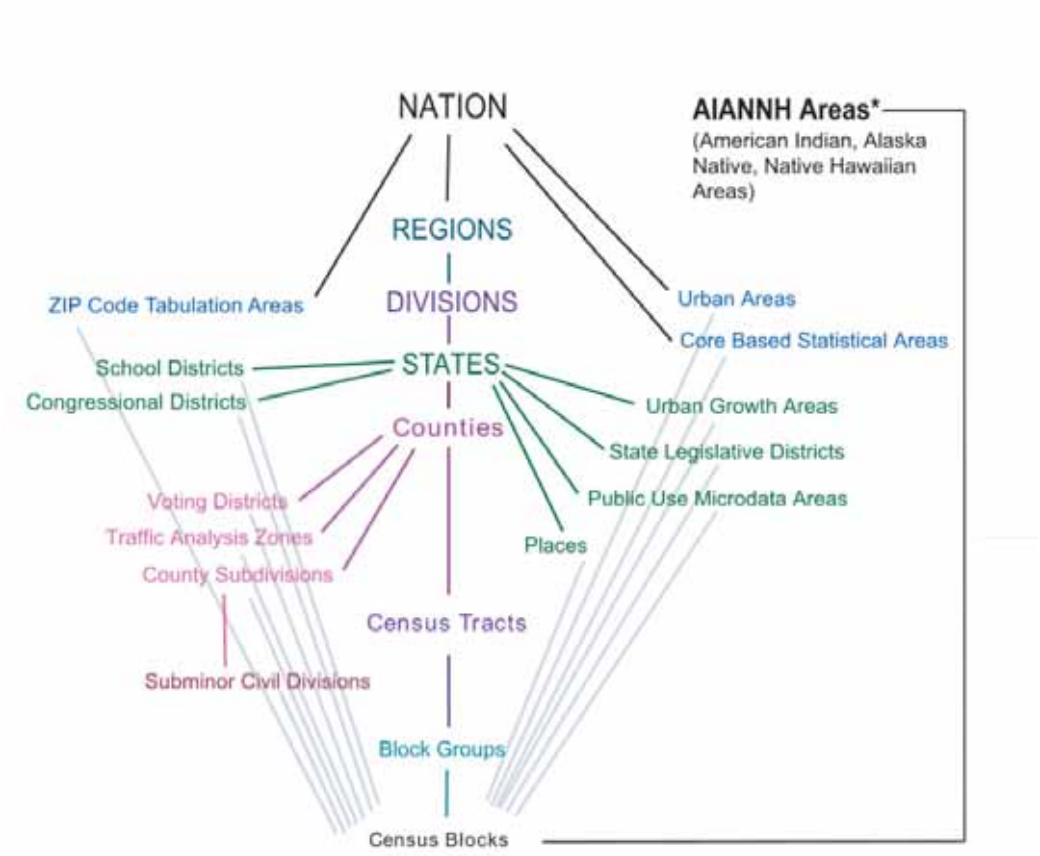


Figure 3-1: The geographical hierarchy used by Census in organizing its various surveys [38].

square mile in states such as Nevada. This diversity in the number of residents and number of households in various regions is one of the reasons Census must work to protect respondent information. In many cases, because of the uniqueness of a given area, it may be possible to identify census respondents. For example, in Figure 3-3 we display graphical representations of the distribution of population and number of households for the country in the form of Violin plots. As can be seen, there is wide variability in both population and number of households even at the census block level. Census blocks are comprised for the most part of roughly several hundred people, but in densely populated areas there are outliers with several thousand people; there is a similar picture for the number of households in a block. Block groups are larger consisting of typically a few thousand

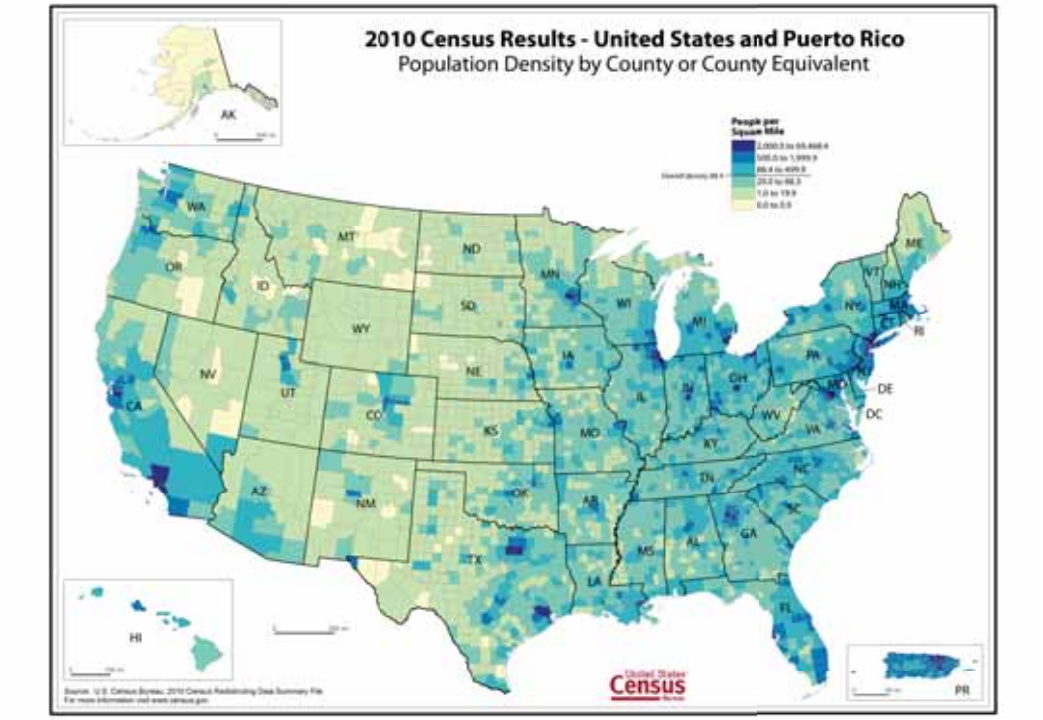
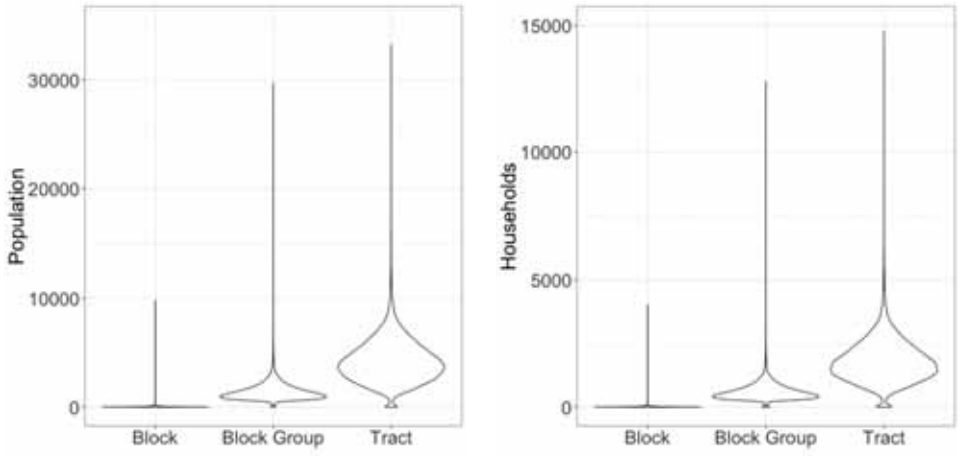
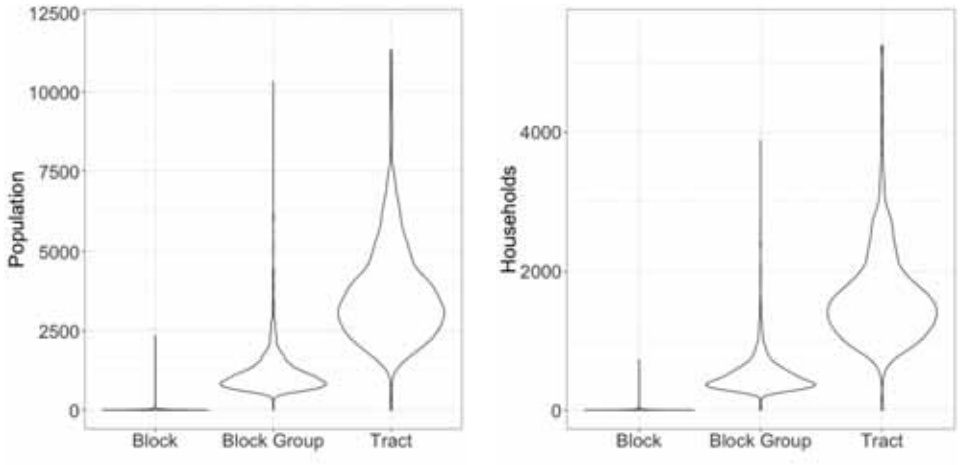


Figure 3-2: Map of population density across the United States from the 2010 census [35].



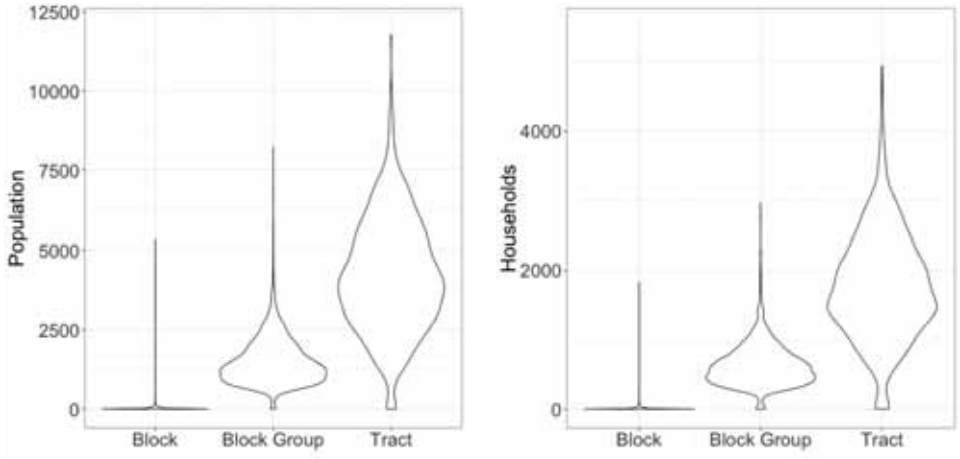
(a) (b)

Figure 3-3: Violin plots of population and households for census tracts, block groups and blocks across the nation.



(a) (b)

Figure 3-4: Violin plots of population and households for census tracts, block groups and blocks in Iowa.



(a) (b)

Figure 3-5: Violin plots of population and households for census tracts, block groups and blocks in Virginia.

people, but here also there is considerable variability. Census tracts may range from population sizes of several hundred in very sparsely populated areas to upwards of 30,000 people. The distribution of population and number of households for blocks, block groups and tracts in a state like Iowa is shown in Figure 3-4. This should be contrasted with the distribution for Virginia shown in Figure 3-5.

Finally it is important to note that census blocks do not always line up with other regions of interest. An important example is the use of census data to determine boundaries of both Congressional and State Legislative districts. Shown in Figure 3-6 are the boundaries for two Congressional districts in Virginia. The boundaries for the districts are shown in black. Census tracts are indicated in purple; census block groups are indicated in orange; and census blocks are indicated in gray. The boundaries for tracts, groups and blocks are quite complex indicative of geography but also complex population patterns. The boundaries of a Congressional district (as well as a state legislative district) are determined through a redistricting process that makes use of the information provided in the PL94 census product (discussed below).

3.2 Census Process and Products

By April 1, 2020 (Census Day) every home will receive a request to participate in the 2020 census. This is the reference data for which respondents report where they usually live. Census then also canvasses group quarters (dorms, etc.) in April. Respondents indicate

- The number of people who live and sleep in a residence most of the time; the homeless are asked to respond as well,
- The ownership status of the household,
- Sex of the residents of the household,
- Age of the residents and their date of birth,
- Whether the residents are of Hispanic origin, [1]
- Race of the residents. This can be any or all of the 63 possible races as designated by the Office of Management and Budget (OMB).

[1] Census refers to this information as the Hispanicity of the respondent.

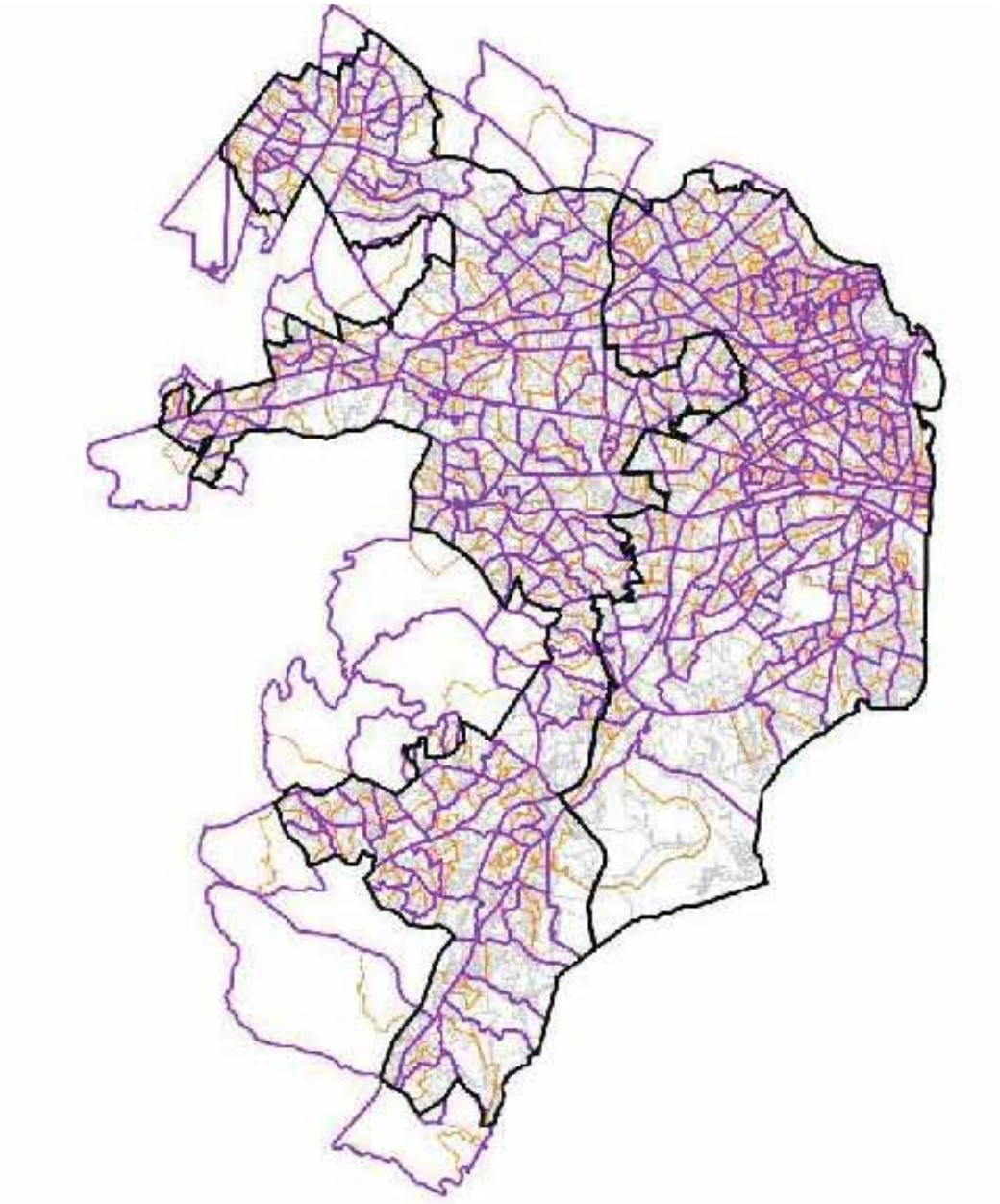


Figure 3-6: A map of two adjoining Congressional districts in Virginia. The black lines indicate the district boundaries; the purple lines indicate boundaries of census tracts; the orange lines indicate boundaries of block groups; the gray lines indicate census blocks.

The 2020 census will also collect information about US citizenship, but respondents will not be asked to indicate their citizenship on the census questionnaire. Instead this will be inferred from existing administrative records (e.g. Social Security Administration, Internal Revenue Service, etc.).

The respondent data are collected into a set of what Census terms microdata, a list of records indicating the responses for each resident. As the responses are received, records are de-duplicated and addresses are validated to insure that every person is counted only once. This forms the Census Unedited File or CUF. Where data are missing or inconsistent the Census employs a process known as imputation and edits the CUF to produce the hundred percent detail file or HDF. The final step is to identify those cells in the various tabular summaries where it may be possible to identify respondents. Here the Census performs confidentiality edits and swaps households as discussed further in Section 3.3. From here the various tabular summaries would be produced.

The Census Bureau through its surveys is responsible for the following products:

Apportionment count Apportionment is the process of dividing the 435 seats of the House of Representative among the states. The count is based on the resident population (both citizen and non-citizen) of the 50 states. An example of the result from the 2010 Census is shown in Figure 3-7 and must be delivered to the President and Congress by December, 2020.

PL94-171 Public law 94-171 directs the Census Bureau to provide redistricting data for the 50 states. This is the first product that must be produced after the apportionment count is complete. Within a year of the 2020 census, the Bureau must send data agreed-upon with the states to redraw state congressional and legislative districts. To meet this requirement the Census has set up a voluntary program that makes it possible for states to receive population estimates as well as racial and Hispanicity distributions for areas relevant to the state congressional and legislative election process. An example of the tables provided in this product is shown

Table 1. APPORTIONMENT POPULATION AND NUMBER OF REPRESENTATIVES, BY STATE: 2010 CENSUS

STATE	APPORTIONMENT POPULATION (APRIL 1, 2010)	NUMBER OF APPORTIONED REPRESENTATIVES BASED ON 2010 CENSUS	CHANGE IN SEATS FROM CENSUS 2000 APPORTIONMENT
Alabama	4,802,982	7	0
Alaska	721,523	1	0
Arizona	6,412,700	9	+1
Arkansas	2,926,229	4	0
California	37,341,989	53	0
Colorado	5,044,930	7	0
Connecticut	3,581,628	5	0
Delaware	900,877	1	0
Florida	18,900,773	27	+2
Georgia	9,727,566	14	+1
Hawaii	1,366,862	2	0
Idaho	1,573,499	2	0
Illinois	12,864,380	18	-1
Indiana	6,501,582	9	0
Iowa	3,053,787	4	-1
Kansas	2,863,813	4	0
Kentucky	4,350,606	6	0
Louisiana	4,553,962	6	-1
Maine	1,333,074	2	0

Source: 2010 Census Apportionment, Table 1

Figure 3-7: A partial list of the apportionment count determining the number of Congressional representatives from each state [39].

in Figure 3-8.

Summary File 1 Census produces a set of demographic profiles after the apportionment and redistricting reports are complete. Summary File 1 (SF1) provides population counts for the 63 OMB race categories and Hispanicity down to the census block level. The report contains data from questions asked of all people and about every housing unit and includes sex, age, race etc. The report consists of 177 population tables, 58 housing tables down to the block level as well as tabulations at the county and tract level. SF1 also provides special tabulations for areas such as metropolitan regions, Congressional districts, school districts etc.

Summary File 2 Summary File 2 (SF2) contains cross-tabulations of information on age, sex, household type, relationship, size for various races as well as Hispanicity down to census tract level as long as the population in the tract exceeds 100 people.

	Virginia	Block 1000, Block Group 1, Census Tract 2001.02, Alexandria city, Virginia	Block 1001, Block Group 1, Census Tract 2001.02, Alexandria city, Virginia
Total:	8,001,024	0	658
Population of one race:	7,767,624	0	630
White alone	5,486,852	0	225
Black or African American alone	1,551,399	0	176
American Indian and Alaska Native alone	29,225	0	3
Asian alone	439,890	0	132
Native Hawaiian and Other Pacific Islander alone	5,980	0	0
Some Other Race alone	254,278	0	94
Two or More Races:	233,400	0	28
Population of two races:	214,276	0	27
White; Black or African American	62,204	0	1
White; American Indian and Alaska Native	25,771	0	0
White; Asian	59,051	0	3
White; Native Hawaiian and Other Pacific Islander	2,618	0	1

Figure 3-8: An example of a population table in the PL94-171 summary file [39].

American Community Survey The American Community Survey (ACS) is an ongoing survey that has taken the place of the decennial long form. It is performed annually. Each year Census contacts 3.5 million households and asks that they fill out a detailed questionnaire. The survey is far more extensive than the decennial census and gathers information about household makeup, type of housing, citi-

zenship, employment etc. The information is used by a variety of stakeholders. Perhaps most importantly, the data are used to guide the disbursement of federal and state funds.

Public Use Microdata Sample Census provides edited samples of the microdata records that make up the decennial census and the ACS. These records are assembled for areas that contain a minimum population of 100,000 (known as PUMAs) and are edited to protect confidentiality. The PUMS provides only a 10% sample of a PUMA.

3.3 The Need for Disclosure Avoidance

It was realized early on that some disclosure avoidance was necessary as the population and housing densities of the United States are not distributed in a homogeneous manner. Owing to special aspects of a location it may be possible to identify the particular person or persons living there. This would constitute a violation of Title 13. For example, Liberty Island, the base of the Statue of Liberty has one household listed, that of the Superintendent of the Monument and his wife [13]. Thus by focusing on this location and using external sources it should be possible to identify the residents of that particular household. For this reason, the information for this location is swapped with that of another household. A history of the methods used in the past 50 years to effect disclosure avoidance is available in the paper by McKenna [24]. We briefly describe these here to provide some context for this report. The discussion below is not complete but illustrates the evolution of the need to offer improved disclosure avoidance.

Long form data Long form census data have never been published at the lowest level of census geography (presently census blocks). The long form data were generally collected as part of the decennial census but in 2010 this data was relegated to what is now called the American Community Survey (ACS) which began

in 2005. The ACS only publishes data down to the block group level.

1970 Census The 1970 Census utilized suppression of whole tables as opposed to suppression of cells. The choice to suppress was based on the number of people in households in a given area. This approach had limitations in that tables with complementary information were not suppressed making it possible in some cases to infer the suppressed information. As indicated by McKenna, cells within an original table could still show an estimate of 1 or 2 people.

1980 Census The 1980 Census retained the approach of the 1970 census but modified it further by now suppressing tables with complementary information and zeroing cells with counts of 1 or 2. However some population counts were not suppressed at any level. In some cases, one could still infer complementary data by subtracting data for various counties from state populations to infer population results for a county that had been suppressed.

1990 Census The 1990 census was the first to employ the concept of swapping. The 100% data (namely PL94, Summary File 1 and Summary File 2) were published down to the block level. But, where there was risk of potential disclosure, a confidentiality edit was performed on the census microdata. For those small blocks deemed at risk, Census selected a small sample of households with a higher sampling rate of such at-risk households used in small census blocks. These at-risk records were paired with other census records from other geographic locations using a set of matching rules. The matching process preserved key attributes such as household size, the age of those residing in a given location, etc. The household records are then swapped and the interchanged version is what is used for the Census Edited File that then forms the source of the various tabular summaries. The rate of swapping is not disclosed so as to prevent possible reverse engineering of the process. In addition, Census began using rounding of entries as well as top and bottom coding to prevent respondent identification arising from

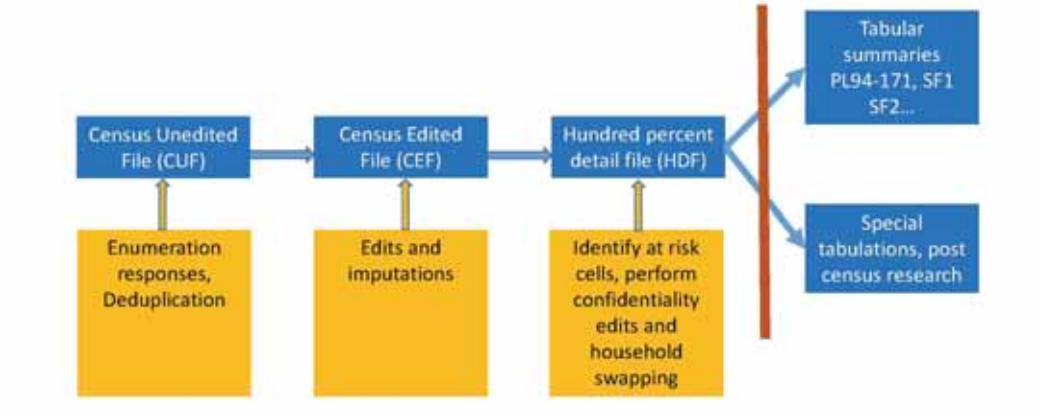


Figure 3-9: A graphical depiction of the disclosure avoidance process used in the recent 2010 census.

age extremes etc.

2000 Census For the 2000 census, more emphasis was given to protecting small blocks and block groups from possible re-identification. For this census, the race category was expanded to include 63 possible alone or combined races. The probability of swapping was increased to those cases where disclosure risk was thought to be higher such as cross-tabulations of key variables, smaller blocks, and also households that contained unique races in that census block.

2010 Census The approach to disclosure avoidance used in 2010 largely followed the approaches developed in the earlier 1990 Census as discussed above. In addition, Census developed partially synthetic data for group quarters in which it blanked values that were assessed as at risk and instead substitutes those values with data obtained from regression models. In summary the disclosure avoidance process follows steps outlined graphically in Figure 3-9. In the next section we discuss why this approach was ultimately judged inadequate.

4 THE CENSUS RE-IDENTIFICATION VULNERABILITY

In this section we discuss the vulnerability discovered by Census using the 2010 census data. We then examine the fundamental basis of the vulnerability: the results demonstrated in 2003 by Dinur and Nissim [5] that releasing an overly large number of statistics about a database allows one to perform reconstruction of the detailed data comprising that database. This result holds true even when one tries to preserve privacy by noising the results of database queries. We verify some of their observations in this section. We also offer a reinterpretation of their results in terms of information theory. Our discussion essentially validates the conclusion of Census that it is possible to reconstruct census microdata even after the application of traditional disclosure avoidance techniques like swapping, top and bottom coding etc.

4.1 Reconstruction of Census Tabular Data

The tabular summaries found in Census products such as PL94-171, SF1 and SF2 have been viewed in the past as safe to publish. These summaries are built using census microdata and it is this microdata that is controlled via disclosure avoidance. For the 2010 census the techniques discussed in Section 3.3 were all used; randomized swapping of households, top and bottom limitations on populations and ages, etc.

In 2018 Census looked at the feasibility that the tabular summaries could be processed to infer the microdata records that were used to produce them [1]. This had not been thought to be feasible owing to the large amount of data and computation involved. Such reconstruction of the microdata is not yet a violation of Title 13 since no personal data (e.g. names, addresses, etc.) are used when these tables are built. But, as in other re-identification attacks, if external data can be joined with the microdata then it may be possible to relink the microdata with

the associated personal data.

In creating the major products published by the Census, each time a cell is populated in a table it is a result of a query made on the microdata. For 2010 the number of queries (or equivalently the number of tabulations) in the PL94 publication is about 3.6B or about 10 for every person in the US. For SF1, the number of tabulations is 22B for population and 4.5B for tabulations of households or group quarters. For SF2 there are 50B tabulations. And for the survey of American Indians and Alaskan Natives there are 75B tabulations. Thus Census publishes a total of 155B queries over the population and households of the US. The population of the US in 2010 was approximately 310M and so many more queries than people (by a significant multiple) have been issued. Most of the microdata entries used to produce these tables have not been processed through traditional disclosure methods.

To test the likelihood of reconstruction Census selected only a subset of the tables that are published. These were

P001	Total population by block,
P006	Total races tallied by block,
P007	Hispanic or Latino origin by race by block,
P009	Hispanic or Latino and not Hispanic or Latino by race by block,
P011	Hispanic or Latino and not Hispanic or Latino by race by age (≥ 18) by block,
P012	Sex by age by block,
P012A-I	Sex by age by block iterated by race,
P014	Sex by age (< 20) by block,
PCT012012A-N	Sex by age by tract iterated by major race alone.

Each table entry is equivalent to an integer-valued linear equation over the microdata tables. For example, if we set the count of people in tract t who are

male and who are 27 years old to $T_{t,M,27}$ then this is tabulated via the equation

$$T_{t,M,27} = \sum_p \sum_r \sum_b B_{p,M,27,r,b}, \tag{4-1}$$

where p sums over the internal person number in the microdata, r sums over the possible races, and b sums over the block codes associated with tract t. The summand B is a selector that is 1 if a record indicates a male of age 27 of any race residing in a block in tract t and zero otherwise [17]. The sum over race is necessary to pick up one of the 63 combinations of race recognized by OMB.

To solve the resulting collection of equations, Census used a state of the art optimization solver known as Gurobi [12]. The Gurobi solver attempts to find the best integer solution to the set of equations corresponding to the tabulations. To break up the problem into manageable pieces Census applied the solver at the tract level. The solver was able to solve the resulting systems with few exceptions. The microdata for the entire US was determined in this way for all 70,000 Census tracts and all 11M Census blocks. To perform the relevant calculations, a virtual parallel cluster was instantiated using Amazon Elastic Cloud facilities and, for this workload and cluster configuration, completed the task in several weeks. Such a task therefore is not outside present day capabilities.

The resulting reconstructed microdata contained

- A geocode at the block level
- A binary variable indicating Hispanic origin (or not) and one of the 63 possible OMB race categories
- Sex
- Age (by year).

Census does publish a sample of the microdata called the Public Use Microdata Sample (PUMS) for use by demographers and other researchers for both the decennial census and for the American Community Survey, but these are rigorously

curated to make sure individual information cannot be inferred. For example, the geographic resolution is limited to areas with populations over 100000. In contrast, the reconstructed data has no population threshold and contains data like single year ages, race, and ethnicity at the block level.

The next step was to see if the reconstructed microdata could then be linked with commercially available marketing data. Some of this data is freely available or could be reconstructed using public records, but more complete and current databases can be licensed through marketing research firms. Such commercial data typically contain names, addresses, sex and birthdate but typically do not contain information regarding race and ethnicity. While not investigated in this case, Census data also contain information about family make-up. Using the reconstructed database, and acquiring commercial data, Census performed a database join using the age, sex and block locations as the common columns of the two datasets. The entries in the resulting table would now have the name and address of the respondent. If correct, these would be a re-identification of the microdata records. Release of this information would constitute a violation of Title 13.

Census determined that 46% of the reconstructed records matched correctly to the internal microdata. If a fuzzy match on age were used, 71% of the records matched. Thus the reconstruction algorithm using only some of the Census tables matched correctly 71% of the US population. Of those internal Census records, 45% were successfully mapped to a corresponding record in a commercial database again using fuzzy age matching with a one year uncertainty. Census then took the records that matched to see if they in turn matched the internal records Census collects when people submit their responses that contain name and address. Of the records that matched the commercial data sets, 39% of these matched exactly with Census records. This corresponds to the successful re-identification of 52M people or 17% of the population in 2010. Previous estimates of the re-identification rate was 0.017% of the population and only 22% of these were confirmed to be correct. The re-identification risk demonstrated by Census is four orders of magnitude larger than had been previously assessed [27].

In section 4.2 we examine a simplified version of this reconstruction problem in which the data set is just a column of bits to verify that the type of attack described above is not specific to the data protected by the Census. It is a general difficulty associated with publishing too many query results about a sensitive dataset.

4.2 Results of Dinur and Nissim

As discussed in Section 4, a key motivation for the development of formal privacy approaches to further secure the 2020 census is the Fundamental Law of Information Recovery. This observation, as quoted by Dwork is that

> "overly accurate estimates of 'too many' statistics is blatantly nonprivate."

By blatantly nonprivate is meant that given some database with information we wish to keep private there exists a methodology to issue queries on the dataset that will allow one to infer a dataset whose elements differ from the original in some number of elements. The number of elements that are not obtained correctly reduces as the size of the database increases. Thus for a large enough database the methodology asymptotically extracts all the elements of the private database.

Dinur and Nissim [5] demonstrated this in a seminal paper by modeling a database as a set of binary numbers whose (private) values we are interested in learning. The database is represented by an array of binary digits:

$$d = (d_1, d_2, \ldots, d_n). \tag{4-2}$$

A *statistical query* is represented by a subset $q \in [1, 2, \ldots, n]$. The exact answer to the query is the sum of all the database entries specified by q:

$$a_q = \sum_{i \in q} d_i. \tag{4-3}$$

An answer $A(q)$ is said to be within ε perturbation if

$$|a_q - A(q)| \leq \varepsilon$$

The *algorithm A* is said to be within ε perturbation if for all the queries $q \subseteq [n]$ the answers A are within ε perturbation. Dinur and Nissim define the notion of $T(n)$ non-privacy if there exists a Turing machine that terminates in $T(n)$ steps so that the probability of determining any fraction of the bits with the exception of a vanishingly small number as the size of the data set increases is essentially one. The result of most relevance to this study is that if the query algorithm provides $o(\sqrt{n})$ perturbation then non-privacy can be achieved with an algorithm that terminates in a number of steps that grows polynomially with increasing data set size. More noise than this is required to get even weak privacy. Dinur and Nissim describe an algorithm using linear programming to demonstrate the existence of such an algorithm. The conclusion is that, even in the presence of noise, a sufficiently capable adversary can infer the secret bits of the dataset. In order to ensure privacy one must restrict the number of queries or add so much noise that the utility of statistical queries on the dataset is potentially degraded.

4.3 JASON Verification of the Dinur-Nissim Results

JASON undertook a verification of the Dinur-Nissim results using a variation of their approach. First we examine the situation where no noise is added to the queries. We then examine the situation where we add noise. We begin by generating a random vector of zeros and ones, $\boldsymbol{d}$, of size n. We then create an $m \times n$ random matrix, Q of zeros and ones. These will be the queries. We then compute the matrix vector product of the query matrix with the database vector. These are the random query results. We then use bounded least squares with constraints to solve the following problem:

$$\text{argmin}\,||Q\boldsymbol{x} - \boldsymbol{d}||^2 \text{ subject to } 0 \leq x_i \leq 1. \tag{4-4}$$

Once this problem is solved we then round the components of the resulting vector

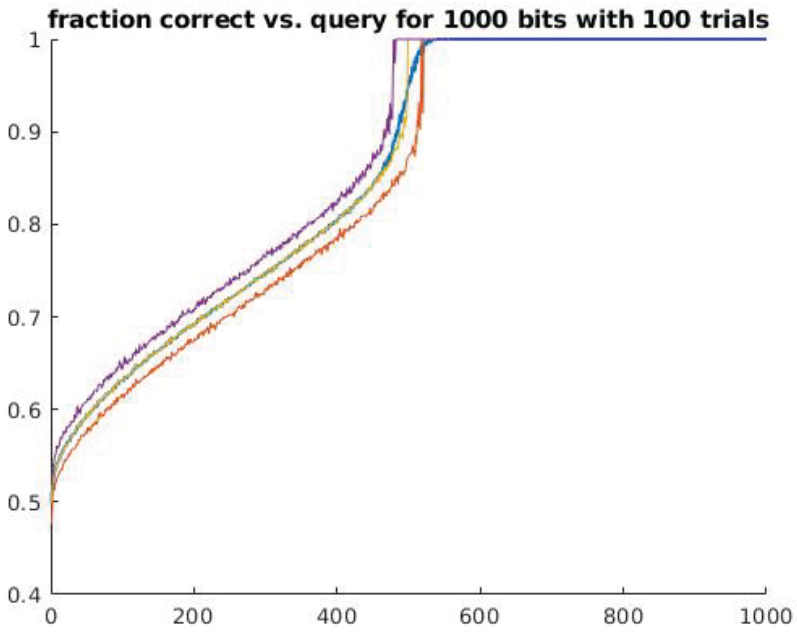


Figure 4-1: Fraction of bits recovered for a 1000 bit Dinur-Nissim dataset as a function of the number of random queries. The lower curve is the minimum fraction recovered, the middle curve is the mean, and the upper curve is the maximum recovered. No noise is added to the query results.

x to 0 or 1. If we issue n queries and our query matrix is not singular,[2] then we would recover the results of the database immediately. But in fact the full database can be recovered with less than n queries in the absence of noise. In Figure 4-2 we plot the fraction of bits computed correctly as a function of the number of queries for a database of size 1000 bits. Because our queries are random we perform 100 trials and plot the 10% decile of the fraction of bits recovered (lower curve), the 90% decile fraction of bits recovered (upper curve) and the mean recovered (middle curve).

With no queries we recover 50% of the bits, but this is of course no better than random guessing. As the number of queries increases we recover more of the bits (although the bits recovered will differ with each random attempt). It is to be expected that we would recover all the bits once we issue 1000 random

[2] singularity would be a very rare event

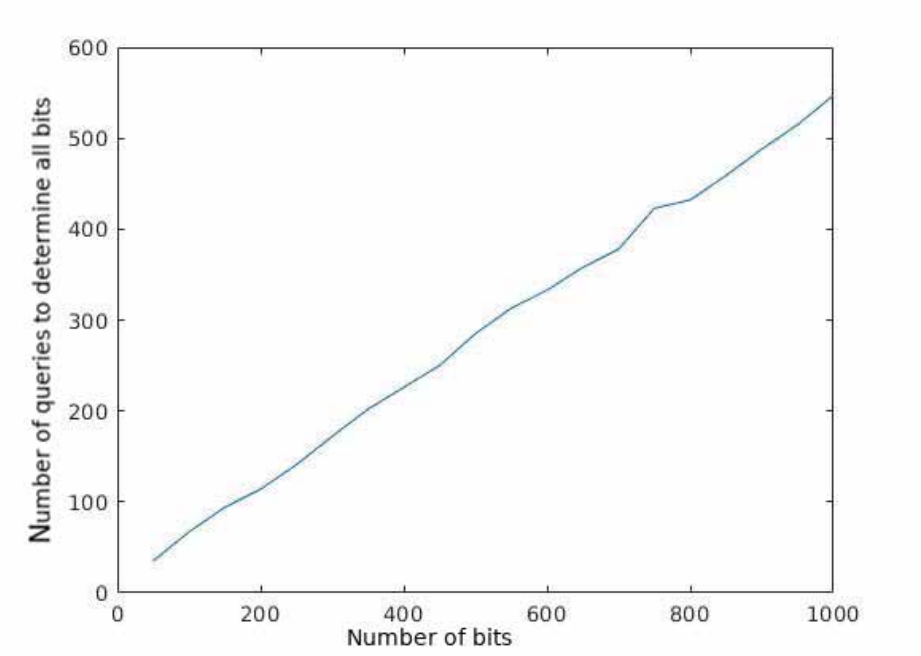


Figure 4-2: Number of queries needed to recover 100% of the private bits in the Dinur-Nissim dataset as a function of the size in bits of the data set.

queries but as is seen in the Figure all the bits are recovered at about the half way mark in the number of queries. If one repeats this calculation for databases of varying size n and asks how the number of queries required to achieve perfect knowledge of the bits varies with n one gets a roughly linear variation in n as shown in Figure 4-2. The slope of this roughly linear variation as a function of increasing database size is shown in Figure 4-3. As can be seen the slope is close to 1/2 indicating that roughly $n/2$ queries are required on average to determine the entire database. This is a special aspect of this particular type of database. A random query response will get information about a number of the bits. For example, if we choose to query two bits at a time by summing the values, then a sum of zero immediately tells us the two bits must be zero. Similarly if we get a sum of 2 we know immediately the two bits we queried must have both been one. Thus one can infer the bits more quickly in a probabilistic sense then simply asking for one bit at a time which would correspond to the query matrix being the identity. In section 4.5 we apply an information-theoretic argument to show that

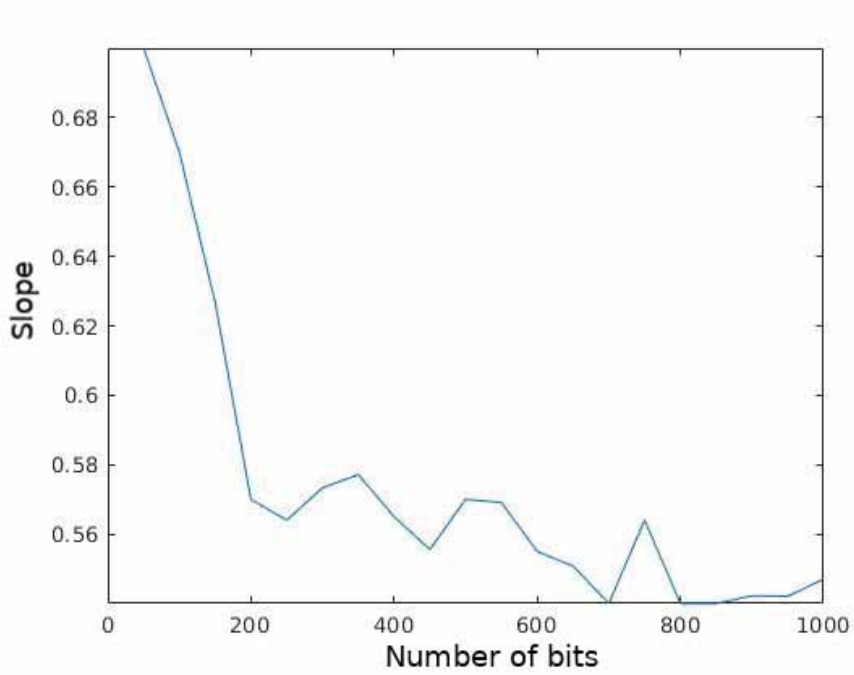


Figure 4-3: Same as Figure 4-2 but each point is normalized by the number of queries. As the number of of bits increases the curve appears to approach a limit of 1/2

the results we get from our least squares approach are not far from optimal.

The results above certainly confirm that, without noise, it is possible through a sequence of queries to infer the entries of a database. It should also be noted that a recovery approach based on optimization will also succeed if one poses more queries than the number of entries in the database. To be sure, the Dinur-Nissim database is special, but it is easily confirmed that through publication of tabular summaries that comprise (sometimes multiple times) the information contained in the database, recovery of the bits, in this case a stand-in for microdata, is possible.

If we think of census data as a (very large) Dinur-Nissim database we can see that the reconstruction attack is quite plausible. In terms of bits, a rough count of the number of bits contained in the Census Edited File might be

- 3 bits to describe the 8 types of group quarters (8 levels),

- 5 bits to describe a person's age (here we assume ages are only reported in intervals of 5)
- 1 bit to describe Hispanic origin (2 levels),
- 6 bits to describe race (63 OMB race designations),
- 24 bits to describe the 11 million census blocks,

for a total of 39 bits per person. If we estimate that in 2010 there were 3×10^8 residents in the US this totals to 1.2×10^{10} bits. If we examine the number of queries in a full cross table this would be

$$(8 \times 20 \times 2 \times 63) \times 1.1 \times 10^7 = 2.2 \times 10^{11}$$

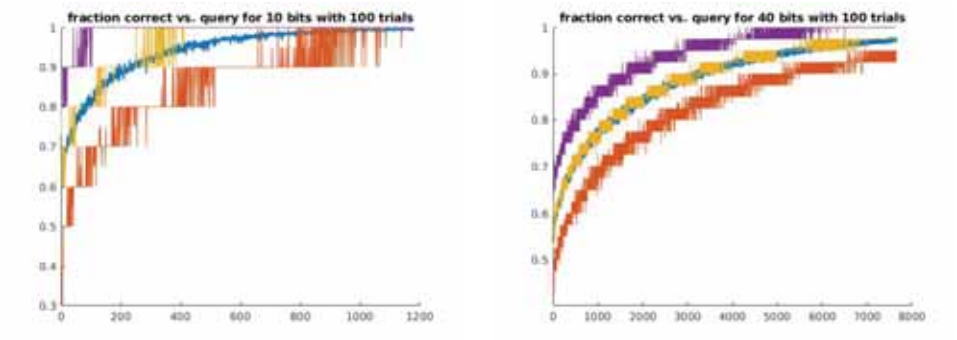
This rough estimate indicates that the census tables "overquery" the data set by a factor of almost 20. If we treat the Census database reconstruction effort as an attempt to infer the bits in a large Dinur-Nissim database there is no question the database (up to the edits that are used to create the tables) could be reproduced with perfect accuracy. A similar argument using the idea of Boolean satisfiability (SAT) solvers is given in [10].

4.4 Queries in the Presence of Noise

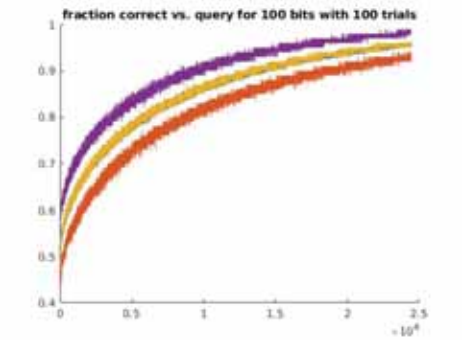
Given the vulnerability discussed above it is perhaps of more interest to examine the number of queries that must be issued to recover the database when each query is perturbed by noise. To examine this, we used the same bounded least squares optimization approach but in the presence of noise. For a dataset size of n bits we added to each random sum a perturbation sampled from a normal distribution of mean 0 and variance $\sqrt{n}\log(n)/2$ where n is again the number of secret bits in the database. The reason for this particular choice was to see if the optimization approach would fail with an increasing number of queries. According to Dinur and Nissim if one adds noise with an amplitude of greater than $O(\sqrt{n})$ then recovery

should be impossible. We were unable to confirm this observation. Instead, as the number of queries increases, an increasing fraction of the correct bits is returned. This is most likely not in conflict with the theorems of Dinur and Nissim as they require that the adversary be time bounded whereas in our approach we do not impose any time limit but instead continually issue queries. The results are shown in Figure 4-4. In the Figure we show the fraction of bits determined correctly as a function of the number of queries for databases of varying size. For each database of size n we added a random perturbation sampled from a normal distribution of mean 0 and variance $\sqrt{n}\log n/2$ to each query.

We perform a query of size m 100 times and provide some statistics for the results. The red, yellow and purple lines indicate the 10%, 50% and 90% deciles respectively of fraction of bits recovered correctly; the blue lines indicate the mean of the fraction of bits recovered correctly. As can be seen, the number of queries required increases greatly, but, in all cases, all metrics measuring the fraction of bits recovered correctly increase towards one. Thus if one is willing to issue a large number of queries, for example, a large multiple of the number of bits, eventually one will learn the internal records of the database. Apparently, the use of random queries will provide results that average out the applied noise and recover the required information. In some ways this is to be expected. For example if we were allowed to issue directly a query for bit i of the n bits in the presence of noise, we would have received a random response, but continual averaging over the responses would have recovered the result regardless of the amount of noise. Indeed we would have predicted that we would have required a number of queries which is some constant factor of the variance. We discuss this further in Section 5 where we consider how many queries are required for a given noise level to recover the internal bits. In the next section we apply information theory to compute idealized estimates of the number of queries required to infer the internal data of the Dinur-Nissim database both in the absence and presence of noise.



(a) (b)



(c)

Figure 4-4: Fraction of bits recovered as a number of queries for databases of size 10, 40, and 100 bits. For each case we infuse the query results with Gaussian noise of means 0 and variance $\sqrt{n}\log n/2$. The red, yellow and purple lines indicate the 10%, 50% and 90% deciles respectively of fraction of bits recovered correctly; the blue lines indicate the mean of fraction of bits recovered correctly. Note that as the number of queries increase, the fraction of bits recovered grows until all the bits are recovered with near certain probability.

4.5 Information Theory and Database Uniqueness

The purpose of this subsection is to look at Dinur & Nissim's [5] fundamental results about database reconstruction from alternative points of view, namely linear algebra and (especially) information theory. The discussion is rather lengthy (but we hope pedagogical) so we have relegated it to an Appendix, but we summarize the main results:

1. In the absence of noise, a database of $n \gg 1$ bits is determined by the re-

sults of approximately $2n/\log_2 n$ queries, on the average over all possible databases. Put differently, we can expect to recover most of the bits of most databases.

2. If noise with variance $\sigma_N^2 < n/48$ is added to the results of each query, the database remains determined by no more than $\sim n$ queries on average.

3. If the noise variance $\sigma_N^2 \gg n/16$, we expect to require $\sim 16\sigma_N^2$ queries to fix the bits uniquely.

It should be noted that there are at least two facets to DN's results: (i) $o(\sqrt{n})$ noise allows the database to be uniquely specified using algebraically (in n) many queries; and (ii) the bits can actually be reconstructed in polynomial time using linear programming. Apart from a few obvious remarks about linear algebra in the noiseless case, we have nothing to say here about the computations required to do the actual reconstruction. Our information-theoretic arguments advanced here are nonconstructive, in much the same way as the Shannon channel-capacity theorem [31], which does not say by what encodings the capacity can be achieved.

5 DIFFERENTIAL PRIVACY

The Census has proposed the use of Differential Privacy (DP) as the basis for its future Disclosure Avoidance System (DAS). The goal of DP is to prevent one from learning about the possible participation of an individual in a survey. The idea is that the result of a query into the dataset provides results that are largely the same even if an individual opted out of participating in the survey. This is accomplished by adding noise to the results of queries so that one cannot easily perform the types of record linkage attacks that have determined the details of database records from queries in the past. DP introduced by Cynthia Dwork [7, 8] and colleagues and developed since then in a vast research literature is viewed as the present gold standard for formal privacy guarantees. The definition is phrased in a language that may be unfamiliar, so we go over it in detail.

The setting is databases and database queries. A database D is a collection of records. Each record has attributes (age, sex, HIV-positive, wealth, or whatever), and each attribute has a range of values it can take. A query is just some function on the database. For instance, "how many records are there", "what is the average age of HIV-positive people", and so forth. We think of attributes being exact and queries giving precise answers, but that is not always desirable as we have discussed previously and is in fact a mental shortcut. Age is reported in years, not days, so people with age 12 are those aged between 12 and 13. Then average age is also reported in years, not some exact number like 62381/129.

DP is a property of algorithms for answering queries. It is clear that, to preserve privacy, queries cannot just return the right answer, so one can think of an algorithm that answers a query as adding noise to the correct answer. Adding noise means that the algorithm is not deterministic, but probabilistic, using random numbers. The approach in which noise is added to the query is known as a mechanism.

An algorithm $\mathcal{A}$ is ε-DP (ε-differentially private) if

$$e^{-\varepsilon} < \frac{\Pr(\mathcal{A}((D)) \in T)}{\Pr(\mathcal{A}(D') \in T)} < e^{\varepsilon}$$

where D and D' are any two databases that differ by one record. The probabilities come from the random numbers that $\mathcal{A}$ uses. T is the set of possible outcomes of $\mathcal{A}$. For instance, if the query was for average age, then T would be an interval like [37,38), meaning that the average age is between 37 and 38. Alternately, if $\mathcal{A}$ returns continuous values, then one needs to measure the probability that the result lies in an interval, rather than takes on a specific value.

A key element of DP is the notion of the privacy budget. In the DP literature this is typically labeled ε. The notation is set up so that a value of $\varepsilon = 0$ indicates zero privacy loss. The technical definition of a DP algorithm is as follows:

Theorem. *An algorithm $\mathcal{A}$ satisfies differential privacy if and only if for any two datasets D and D' that differ in only one record, we have that for all results T that lie in the range of the algorithm $\mathcal{A}$*

$$Pr[\mathcal{A}(D) \in T] \leq \exp(\varepsilon) Pr[\mathcal{A}(D') \in T].$$

Equivalently the ratio of probabilities

$$\frac{Pr[\mathcal{A}(D) \in T]}{Pr(\mathcal{A}(D') \in T} \leq \exp(\varepsilon).$$

Note that there is nothing special about D and D' so we can write the inequality in a symmetric two-sided manner as we did above:

$$\exp(-\varepsilon) \leq \frac{\Pr[\mathcal{A}(D) \in T]}{\Pr[\mathcal{A}(D') \in T]} \leq \exp(\varepsilon).$$

If an algorithm satisfies the definition of being differentially private, the expression above provides a bound on how much additional information one can infer from adding or deleting a record in a database. This will prevent learning about a specific record through the examination of the two datasets for example through database differencing. It also makes record linkage attacks more difficult in that it introduces uncertainty in the query results.

Perhaps of more importance, DP algorithms by definition provide formal bounds on how many queries can be made before the probability of learning something specific about a database increases to an unacceptable level. This is the real role of the privacy budget. A DP algorithm with a large value of ε indicates that the ratio of probabilities of learning a specific result in two datasets with one record differing is large and so implying that the query using the algorithm discriminates strongly between the two datasets. On the other hand, a small value of ε means little additional information regarding the dataset is learned. It is not hard to show that DP has several properties that make it possible to reason about how the privacy budget is affected by queries.

Sequential access to the private data degrades privacy Suppose we have an algorithm $\mathcal{A}_1$ that satisfies DP with privacy loss parameter ε_1 and another algorithm $\mathcal{A}_2$ that has a privacy loss parameter ε_2. If both algorithms are composed then the privacy loss parameter for the composed algorithm is the sum of the individual privacy loss parameters. we have

$$\begin{aligned}\Pr[\mathcal{A}_2(\mathcal{A}1(D),D)=t] &= \sum_{s\in S}\Pr[\mathcal{A}_1(D)=s]\Pr[\mathcal{A}_2(s,D)=t]\\ &\leq \sum_{s\in S}\exp(\varepsilon_1)\Pr[\mathcal{A}_1(D')=s]\exp(\varepsilon_2)\Pr[\mathcal{A}_2(s,D')=t]\\ &\leq \exp(\varepsilon_1+\varepsilon_2)Pr[\mathcal{A}_2(\mathcal{A}_1(D')D')=t].\end{aligned}$$

In general, if one composes this way k times the effective ε becomes

$$\varepsilon = \varepsilon_1+\varepsilon_2+\cdots\varepsilon_k.$$

This implies that one must account for all the operations to be performed on the data in order to ensure a global level of privacy over the whole dataset. It also demonstrates, at least in terms of bounds, the cost of a number of queries on a database in terms of overall privacy and that repeated queries on the data will boost the ratio of probabilities. This provides a useful quantitative aspect to assessing disclosure risk atlhough it is not explicitly a statement about disclosure risk.

The privacy budget behaves gracefully under post-processing If an algorithm $\mathcal{A}_1$ satisfies DP with a privacy budget of ε, then for any other algorithm $\mathcal{A}_2$ which post-processes the data generated by $\mathcal{A}_1$, the composition of $\mathcal{A}_2$ with $\mathcal{A}_1$ satisfies DP with the same privacy budget. To see this, suppose S is the range of the algorithm $\mathcal{A}_1$. Then we have

$$\begin{aligned}\Pr[\mathcal{A}_2(\mathcal{A}_1(D)) = t] &= \sum_{s\in S} \Pr[\mathcal{A}_1(D) = s]\Pr[\mathcal{A}_2(s) = t] \\ &\leq \sum_{s\in S} \exp(\varepsilon) Pr[\mathcal{A}_1(D') = s]\Pr[\mathcal{A}_2(s) = t] \\ &\leq \exp(\varepsilon)\Pr[\mathcal{A}_2(\mathcal{A}_1(D')) = t].\end{aligned}$$

It is important in this argument that only the algorithm $\mathcal{A}_1$ accesses the private data of the database. This composition property is quite powerful. One of its most important applications is that if you transform the database into another database with synthetic data processed through a DP algorithm then additional processing of that data will preserve differential privacy. Thus one can create a dataset from the original dataset and preserve differential privacy for future processing of the synthetic data. This feature is an important component of the disclosure avoidance system currently under consideration by Census.

Parallel composition If one deterministically partitions a database into separate parts then one can control the privacy loss. If $\mathcal{A}_1, \mathcal{A}_2 \ldots, \mathcal{A}_k$ are algorithms that respectively only access the (nonoverlapping) partitions of the database D_1, D_2, $\ldots$ D_k then publishing the results of the queries $\mathcal{A}_1(D_1), \mathcal{A}_2(D_2), \ldots \mathcal{A}_k(D_k)$ will satisfy DP but with an ε given by

$$\varepsilon = \max(\varepsilon_1, \varepsilon_2, \varepsilon_k).$$

Such results show that the production of a histogram where the data is partitioned into categories and then counts are published for each category can still preserve a given privacy budget.

5.1 Mechanisms

The definition of DP does not guarantee that there are any DP algorithms, but of course there are. In general, a *mechanism* is a way of generating DP algorithms from data base queries. We discuss some of these below.

5.1.1 Laplace mechanism

Consider a query whose correct answer is some continuous numeric value. The query has sensitivity Δ if the correct answer on any two neighboring databases D, D' can differ by at most Δ. Then an ε-DP algorithm for this query would add $\text{Lap}(\Delta/\varepsilon)$ noise sampled from a Laplace probability distribution to the correct answer, where Lap is the two-sided Laplace distribution. The probability density for the Laplace distribution with parameter β is

$$\frac{1}{2\beta}\exp\left(-|x|/\beta\right).$$

More usefully, to generate a random Laplace variate from a uniformly distributed p between 0 and 1, one can compute

$$\beta\ \text{sgn}(p-0.5)\ln(1-2|p-0.5|).$$

This density has mean 0 and a variance of $2\beta^2$ and is displayed in Figure 5-1. In applications to DP we use the relation $\beta = 1/\varepsilon$. Thus small values of privacy loss imply large values of β and so very broad distributions with large variances. Note that the use of the Laplace mechanism and the associated Laplace distribution matches exactly with the definitions of DP in terms of the bounds on probabilities. Other distributions can be used, for example, a normal distribution, but in this case there may be small violations of the DP bounds for extreme values. A slightly modified definition of DP is required to handle this case but its use would not affect our conclusions so we won't discuss it further.

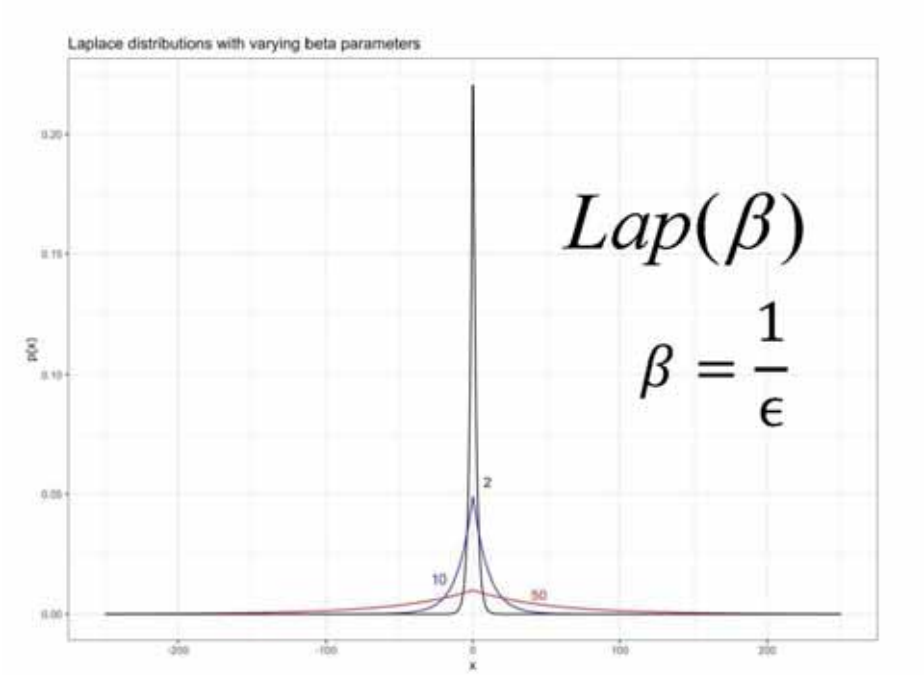


Figure 5-1: The Laplace distribution for several values of the parameter β. A large β corresponds to broad tails.

5.1.2 Geometric mechanism

The Laplace mechanism does not produce integers for integer-valued attributes. The Geometric mechanism adds an integer to the correct answer, where the integer is randomly chosen from a suitable geometric distribution One could instead use the Laplace mechanism and round, but these results are slightly different. The (two-sided) geometric distribution with parameter α has probability density

$$\frac{\alpha-1}{\alpha+1}\alpha^{-|x|}$$

for producing integer x. If Δ is the sensitivity of the query, ε-DP is the same as $\alpha = \exp(\varepsilon/\Delta)$.

5.1.3 Matrix mechanism

In applying DP to the census tables one approach would be to make one colossal query of the confidential data that produces at once all the tables that the public will be able to see. Each number in each of these tables is a count, so the colossal query can be represented as a big matrix M applied to a huge vector $\boldsymbol{c}$ of the confidential data. DP would add noise to each count in $M\boldsymbol{c}$. But this may introduce more noise than is strictly required. A way to deal with this is known as the matrix mechanism [25, 19]. The public tables published by the Census are counts over discrete categories. The (confidential) data is a data base where each record has some attributes, and each attribute only takes on a finite set of values. These include age (from 0 to some upper bound), sex, Hispanicity, race (63 values), and so forth. An equivalent way of representing the data is as a (long) histogram, with one count for each possible combination of attributes. So there would be a count for 'male black-asian hispanics of age 37' and one for 'female white non-hispanics of age 12', and so forth. If these are arranged in some arbitrary order, we can think of the data base as a vector of counts $(x_1, x_2, \cdots, x_n)$. Then the result of a count query (e.g., 'male native-americans') is the inner product $w \cdot x$ where w is a vector of 0s and 1s of length n, with 1s exactly for those places in the histogram that count male native-Americans. This inner product is one of the counts in the publicly released tables. The set of queries that produce all these counts can be represented as the rows of a very large matrix W.

The idea of the matrix method is to answer all these queries (or this one giant query) in two stages. First answer a set of *strategy* queries in a differentially private way, and then combine the answers to these queries to get the queries we want ($W\boldsymbol{x}$). The strategy queries can be represented by some matrix A, one computes $\boldsymbol{m} = A\boldsymbol{x} + \Lambda$, where Λ is a vector of noise chosen so that the result is ε-DP. Then any post-processing of m does not affect privacy, so if $W = UA$, then $W\boldsymbol{x} = U\boldsymbol{m}$, which are the tables we want. One can attempt to find such an A that minimizes the mean error in the output. The process is illustrated graphically in

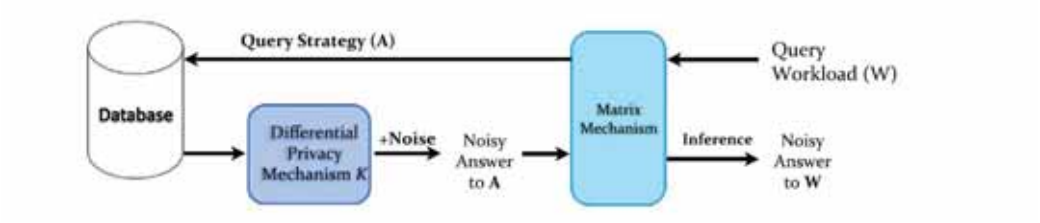


Figure 5-2: Process utilized by the matrix mechanism (from [25]).

Figure 5-2. This is a substantial computation described in the referenced papers.

5.2 Some Surprising Results in Applying Differential Privacy

The definition of DP does not immediately speak to the kinds of errors introduced. Nor does it guarantee that a query has a satisfactory (or any) DP algorithm. Below are presented some examples that indicate that one must be careful sometimes with the result of DP calculations to ensure statistical utility of the results.

5.2.1 Cumulative distribution functions

In [26] an example is given of how DP can affect common statistical measures. For example if we want to compute a cumulative distribution function (CDF) of incomes in some region we would count the number of income values less than some prescribed value and then divide by the total number of incomes to get a distribution. Under DP each time such a query is issued noise is added to the result. Depending on the level of noise injected the resulting CDF may become non-monotonic, something that is mathematically forbidden. Some results are shown in Figure 5-3 for a sample CFD under various values of ε. As ε is increased the generated CFD will converge to the smooth case without noise. The examples shown with a large amount of injected noise could not for example be reliably differenced to provide probabilities over small intervals. This is in fact the point - we cannot focus too clearly on the small scales. The issue identified here can be easily fixed by re-sorting the data so that a monotonic CFD results. The main

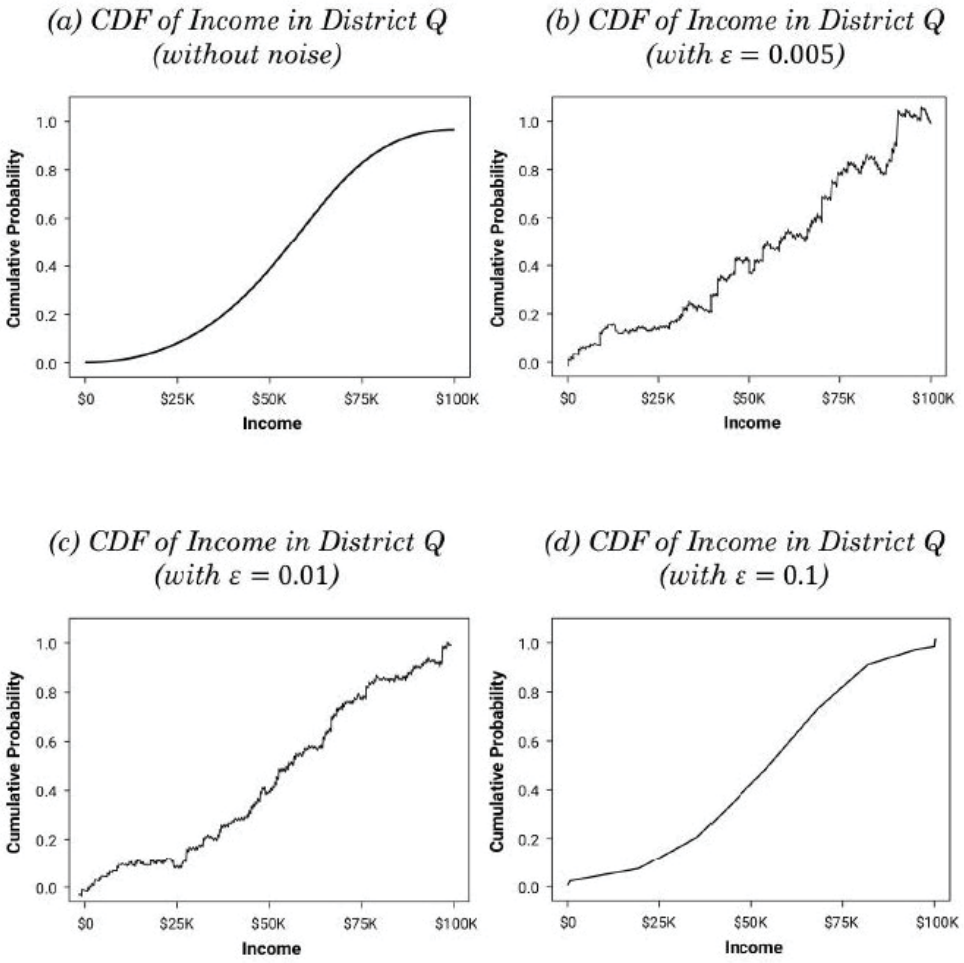


Figure 5-3: An example of a CDF of incomes under various values of the privacy loss parameter (from [26]).

point here is simply to point out possible issues with results published directly under DP.

5.2.2 Median

The examples of mechanisms so far involve additive noise, but the definition does not mention the type of noise. Consider a query that asks for the median. If the middle three elements in the larger database are 0.12, 0.14, 0.19, then if the size

of the database is odd, the median is 0.14, otherwise some tie-breaking algorithm would be needed. The smaller database is the result of removing one record from the larger database. If the number removed is no more than 0.12, the new exact median will be between 0.12 and 0.19. If the number removed is 0.19 or more, the same is true, and if 0.14 is removed, it is also true. So a privacy algorithm could choose any number between 0.12 and 0.19. Note that this algorithm decides what to do based on the data. It satisfies the intuition behind DP in that the result is independent of which record is removed from the database. However, it is *not* ε-DP for any ε. To see that, consider what the algorithm returns for the smaller database, if 0.12 were returned. Then the middle 3 might be 0.10, 0.14, 0.19, and the algorithm could return any value between 0.10 and 0.19. In particular there is a positive probability of returning a value in the interval [0.10, 0.12] for the smaller database, but that's impossible for the larger. So the ratio of probabilities in the definition of DP would be 0, which is impossible for any ε.

For the median, however, the sensitivity Δ is large. If the attribute takes on values between 0 and 1, and in the smaller database half of them are 0 and half of them are 1, then the median for the larger database is whatever value was removed, so $\Delta = 1/2$ (assuming that the algorithm chooses the midpoint for even sized databases). The Laplace mechanism doesn't look at the data, so it will add $\text{Lap}(1/2\varepsilon)$ noise. Answers that then fall outside [0,1] presumably would be truncated to be in range, so there is a positive probability of getting 0 or 1, which will almost always be silly and completely uninformative.

There is a similar story for any quantile, or the min, or the max, but the median is often used as a robust measure of location. Dwork and Lei [6] give a different algorithm that should be generally more satisfactory, but is data-dependent, and can fail (returning $\perp$ (null) in the language of computer science) on weird databases, such as the one in this example.

The decennial census data is just counts, so the peculiarities of medians are not directly relevant, but other statistical agencies and other statistical products

might not be so lucky.

5.2.3 Common mechanisms can give strange results for small n

Another mechanism is known as the random or uniform mechanism (UM). For a query that has a finite range, the random mechanism just chooses one uniformly; For example for the range of integers 0 through 10, choose a query response with probability 1/11. The random mechanism is ε-DP for any ε. If one were to propose a mechanism for a query associated with this finite collection of integers, it would seem undesirable for it to give the correct answer less frequently than the random mechanism does. That is, there may be many DP algorithms for the query, and it is unsatisfactory to chose one whose accuracy (meaning the chance of getting the right answer) is less than just choosing a result at random. For small n, both the truncated Laplace or Geometric mechanisms are unsatisfactory in this way.

There are various mechanisms for producing DP count data, The simplest way to think about these is to assume the data base has records with one sensitive field that has value 0 or 1. Suppose the query that counts the number of 1s needs to be protected. We know the answer is in the range [0, n], so the mechanism needs to produce a value in that range. The Range Restricted Geometric Mechanism (GM) produces

$$\min(n, \max(0, a + \delta))$$

where a is the true answer and δ is an integer chosen (at random) from a geometric distribution

$$(1-\alpha)^{|\delta|}/(1+\alpha)$$

where $\alpha = \exp(-\varepsilon)$ and ε is the parameter in differential privacy. Unfortunately, in this case, 0 and n will be over-represented. Worse, for most probability distributions on a, the actual count, if n is 2, the true answer of 1 is less likely than either of the incorrect answers 0 or 2. This is clearly a small n phenomenon,

but for small and modest-size n the usual mechanisms with various common loss functions give counter-intuitive results (cf. e.g. [4]).

Any mechanism for this problem is characterized by a (column) stochastic matrix P, where $P_{i,j}$ is $\Pr(i|j)$, the probability the mechanism returns i when the true result is j. P is an $(n+1)\times(n+1)$ matrix. The uniform or random mechanism (UM) has $P_{ij,} = 1/(n+1)$, that is, choose any answer at random. The set of all mechanisms can be defined by linear equations and inequalities. The only unobvious one, differential privacy, is expressed by

$$P_{i,j} \geq \alpha P_{i,j+1}, \quad P_{i,j+1} \geq \alpha P_{i,j}$$

for all i and j. The choice of a mechanism then comes down to minimizing some loss function over this polytope, preferably by linear programming. There are n^2 variables and a quadratic number of constraints.

Cormode's paper [4] notes that one can add a number of intuitively desirable constraints on the mechanism by adding linear constraints to this formulation. For instance, one might like the probability the mechanism returns the correct answer to be at least as large as the chance UM returns it, $P_{i,i} \geq 1/(n+1)$. Interchanging the values 0 and 1 in the statement of the problem converts a true answer a into $n-a$. One would expect the mechanism to be oblivious to this choice, which imposes a symmetry contstraint $P_{i,j} = P_{n-i,n-j}$. One would like the correct answer to be at least as probable as any other. The geometric mechanism (GM) satisfies these only for sufficiently large n, at least $2\alpha/(1-\alpha)$, which is roughly $2/\varepsilon$. If one adds the condition that answers closer to the true answer should be more likely than answers further away, then GM requires $\alpha < 1/2$.

For completeness, here is the explicitly fair mechanism of [4], which looks more complicated than it is, and satisfies their various sensible conditions:

$$P_{i,j} = \begin{cases} y\alpha^{|i-j|}, & \text{if } |i-j| < \min(j, nj) \\ y\alpha^{\lceil \frac{|i-j|+\min(j,n-j)}{2} \rceil} & \text{otherwise} \end{cases}$$

where

$$y = \frac{1-\alpha}{1+\alpha-2\alpha^{n/2+1}},$$

so the probability of returning the correct answer is a little larger than in the geometric mechanism, and the probabilities drop off more slowly with distance from the correct answer. The paper gives rules for choosing between this mechanism and GM.

5.2.4 Nearly equivalent queries with vastly different results

Suppose we have a database for which HIV-status is an attribute, with the values 0 or 1. The query might be "are more than half of the records 1?" One sensible way of answering this question using counts would be to ask for the size of the database n, and the number of ones, x, and look at the result. The returned values would have Laplacian or Geometric noise added to them, but unless the number of ones is very near 50%, the answer to the original question just pops out. A different computation, equivalent if exact results are returned, would be to ask if the median value of HIV-status is 0 or 1. As we have seen there is a positive chance of getting a meaningless answer regardless of how different the counts of zeros and ones. A more sensible query would be to ask for the average. The average is not a count query, but it has sensitivity $1/n$ for values between 0 and 1. So a DP query would answer with $\mathrm{Lap}(1/n\varepsilon)$ noise added to the exact answer. This error drops rapidly with increasing n.

5.3 Invariants

The main promise of DP is to limit the knowledge that can be gained by adding or subtracting a record from a database. Informally if we make a small change in the input data the result of the output also undergoes a small change. That this is not always the case has been shown repeatedly through linkage attacks and database differencing. However, if certain results in a database must be openly

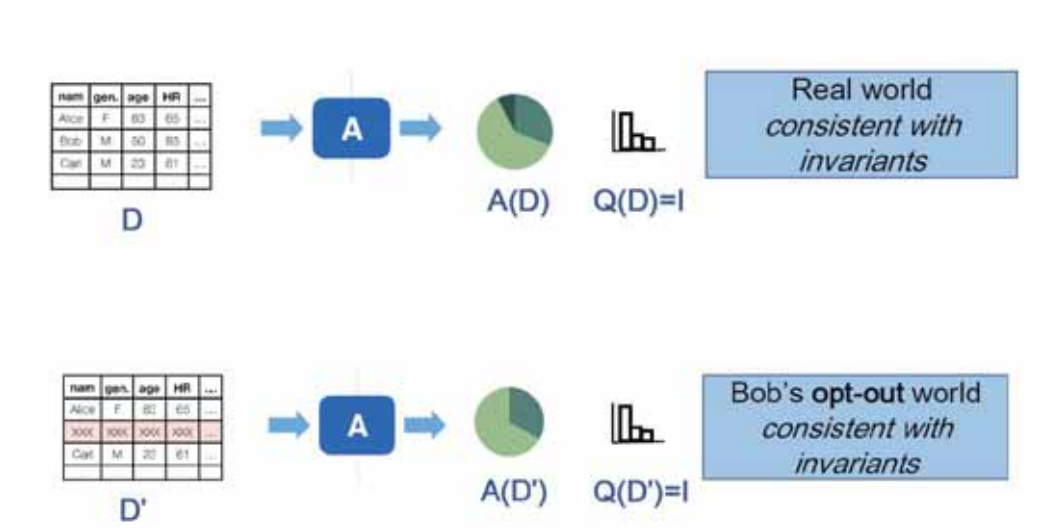


Figure 5-4: DP with invariants must be interpreted relative to a world in which respondents opt-out but consistent with invariants [21].

published without any protection then a small change in the input can have large consequence on the output if the output is directly tied to the small change.

An important example is the notion of an invariant. A simple example of an invariant relevant to the census is the need to publish an accurate count of the population of each state. For the 2020 Census, as in previous censuses, there are plans to publish state populations as exactly as possible and certainly without noise and so the state populations are invariants. In theory, releasing a true count is technically a complete violation of the DP guarantee. This is simply because removing one entry changes the population and so it is immediately obvious that a record has been removed even though we may not know which record.

As briefed to JASON by Prof. A Machanavajhala [21], it is possible to construct various scenarios where releasing an invariant could allow one to infer additional protected information regarding a record. There is to date no worst case characterization of privacy loss in this situation. At best, one can consider the incremental loss in releasing DP results in the presence of invariants. The situation is shown graphically in Figure 5-4. At present, it is not clear to what extent the

addition of invariants constitutes a vulnerability for Census data. As will be discussed below there are many more constraints that lead to invariants than just the population of the states. JASON does not know of a systematic approach to assess this except to perform a risk assessment by attempting to identify DP microdata as was orginally performed by Census in first identifying the existing vulnerability in the absence of noise. We discuss this further in Section 7.

5.4 Database Joins under Differential Privacy

In creating the various Census products such as SF1, the tables are produced through a join between two databases. One contains information about persons and the other about households. Queries such as the number of men living in a particular Census block requires only access to the person database while queries such as the number of occupied houses in a Census block requires only access to the household database. But if one wants to know how many children live in houses headed by a single man this requires a join of the two databases. Joins under DP can be problematic because one must examine the full consequences of removing a record in one table as it is linked to potentially multiple records in other tables. One way to address this is to create synthetic data as the Census is doing for both tables and then perform the join as usual. This however has been shown to produce high error in the results of queries essentially because too much noise is added for DP protection. A number of groups have researched this issue and provided possible solutions. The state of the art is a system called PrivSQL [15] which makes it possible to more efficiently produce tables via SQL commands while attempting to enforce a given privacy budget and while also attempting to optimize query accuracy. An architecture diagram for this system is shown in Figure 5-5. The system must generate a set of differentially private views for a set of preset queries. A sensitivity analysis must be performed and a set of protected synopses are then generated that can be publicly viewed. Census will perform the appropriate queries and create the protected tables using this

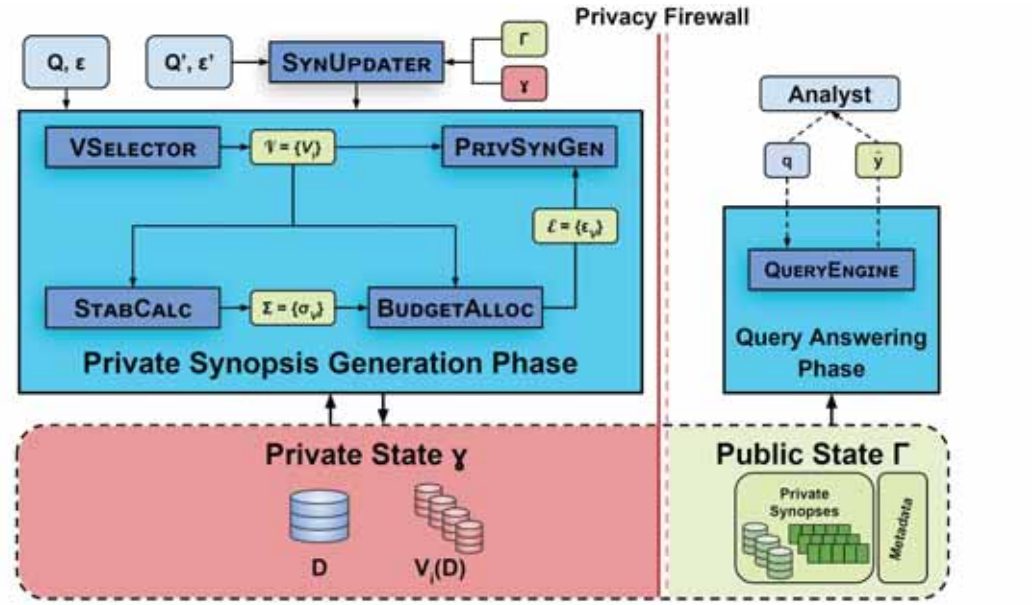


Figure 5-5: Architecture diagram for private SQL queries [15].

approach. Microdata associated with these tables will then be produced. This is at present work in progress, At the time Census briefed JASON their plan was to release a modified version of SF1 but tables requiring the linkage of data from person and housing records could not yet be constructed. It is expected that with further work using PrivSQL it should be possible to eventually produce many if not all of the traditional Census products.

5.5 The Dinur-Nissim Database under Differential Privacy

We provide here an example of the use of methods like DP as applied to queries of the Dinur-Nissim dataset. As discussed in Section 4.2 Dinur and Nissim made use of a simple database consisting of binary numbers to put forth what is now known as the Fundamental Law of Information Recovery, namely, that even in the presence of noise one can determine the contents of a private database by issuing and receiving the responses to too many queries. Here we illustrate that, despite the addition of noise, it is still possible to obtain meaningful statistical information from the database. We create a DN database as an array of randomly chosen

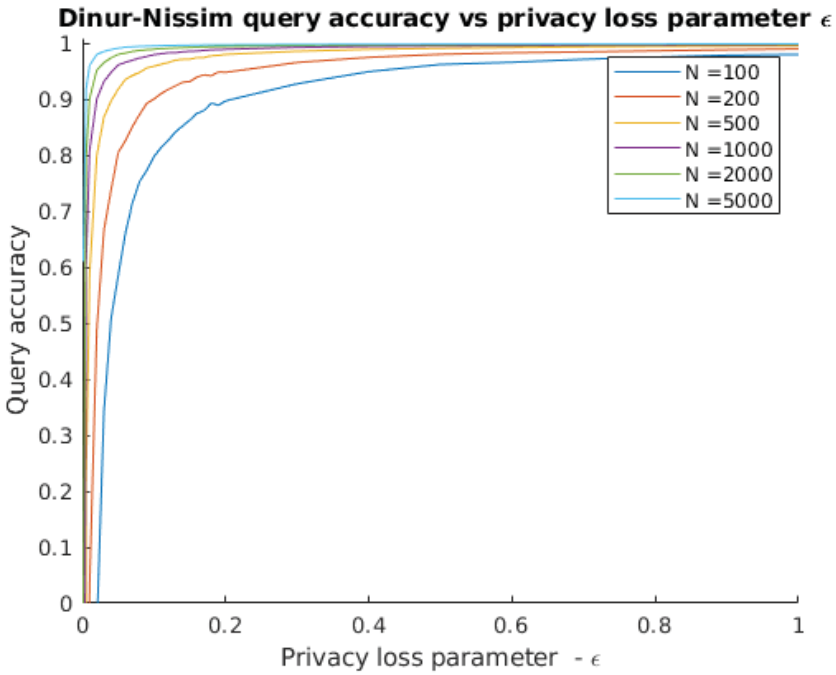


Figure 5-6: Accuracy of a sum query on the DN database. The values of N shown indicate the size of the database.

bits of size N bits. These could be the answer to a survey where the response is yes or no. We would like for example to know how many people responded yes to our survey. The result of our query is just the sum of the bits giving us the number of affirmative answers. For any query of this type issued we add a random amount of noise sampled from a Laplace distribution $\mathrm{Lap}(1/\varepsilon)$ with mean zero and variance $2/\varepsilon^2$. To measure the impact of the additional noise we calculate the query accuracy defined by

$$A = 1 - \frac{|\tilde{S} - S|}{S}$$

where $\tilde{S}$ is the noised sum and S is the sum in the absence of noise. A varies from 1 (no error) and then decreases towards zero and can become negative. Clearly, A of zero is of no utility. For each value of ε and N the number of bits we repeated the calculation 1000 times and reported the average A. The results are shown in Figure 5-6.

As can be seen, the accuracy of a query perturbed using the Laplace mech-

anism depends on the size of the data set. For the smallest dataset of size 100, a privacy loss value of $\varepsilon = 2$ degrades the query accuracy by about 15%. As N is increased the query accuracy improves and for $N = 5000$ the effect of the perturbation due to DP is imperceptible. In fact it would be smaller in this case than the statistical uncertainty associated with the query which varies as $1/\sqrt{N}$. For smaller values of ε the impact of the perturbation becomes more noticeable with the conclusion that smaller values of ε that provide increased privacy protection will not disturb statistical accuracy provided one deals with large datasets.

5.6 Multiple Query Vulnerability

As discussed in section 4 for the Dinur-Nissim dataset, it is still possible to recover the bits of the dataset provided enough queries are issued and optimization is used to get a "best fit" to the bit values. This works in our case even in the presence of arbitrarily large noise. The optimization technique, in our case least squares with constraints followed by rounding, can apparently return a result that converges to the true answer - the values of the bits in the dataset. We note that the residual norm of the optimization in this case will be very large, indicating that when the optimized result is used to compute the right hand side of the linear system representing the queries, the difference with the right hand side presented to the optimizer is very large. This is to be expected as we constrain the lower and upper bounds of the solution to be zero and one respectively. When we apply, for example, Laplace noise to the right hand side, we perturb it so that in some cases it would be impossible for a series of zeros and ones to sum to the indicated right hand side values. The larger is the noise amplitude, the more likely this is to occur. Nevertheless the optimizer will find solutions (effectively averaging out the applied noise) and as the number of random queries is increased the percentage of recovered bits increases.

To put this observation into the context of the Census vulnerability, we generate a Dinur-Nissim database consisting of 4000 randomly chosen bits. We then

generate a query matrix Q of size $N_Q \times n$ where n is the size of the database and N_Q is the number of issued random queries. In this case we set N_Q to be a multiple of the dataset size as this seemed more relevant to the issue faced by Census. That is, given a population, how many queries expressed as a multiple of the population suffice to infer the microdata. In the case of the Dinur-Nissim dataset, it is possible to ask this question even in the presence of noise and, empirically, while the number of queries required to determine the bits does increase with the size of the dataset, eventually, with high probability, all the bits can be recovered.

Given a query matrix and the dataset we compute the matrix-vector product and then set a value of the privacy loss parameter ε (in our case ranging from 0.01 to 1) and added to each component of the vector a random amount of noise sampled from the Laplace distribution. We then applied constrained least squares optimization and examined the fraction of bits recovered correctly. We assume that different bit locations are recovered correctly in computing the fraction recovered, but privacy concerns would certainly arise if the fraction of bits recovered exceeded 0.9. After some number of queries the algorithm succeeds in determining all the bits every time. A Matlab code performing this computation is included in Appendix B.

The results of our experiment are shown in Figure 5-7. Note that if one just guesses randomly, it is possible to recover 50% of the bits and so the minimum fraction of bits recovered is 0.5. The x-axis of the plot (labeled "Query multiple") indicates the number of queries scaled as a multiple of the size of the data set. In this case a multiple of 20 indicates 80000 random queries were made. The y axis indicates the privacy loss parameter. It can be seen that for example for $\varepsilon = 0.01$ and 4000 queries the results are not much better than random. But as the number of queries increases the fraction of bits recovered also increases. As the privacy parameter increases, and the number of query multiples increases eventually all the bits are recovered. This behavior is in line with the results of DP. Not only must one noise the data, one must also restrict the number of queries.

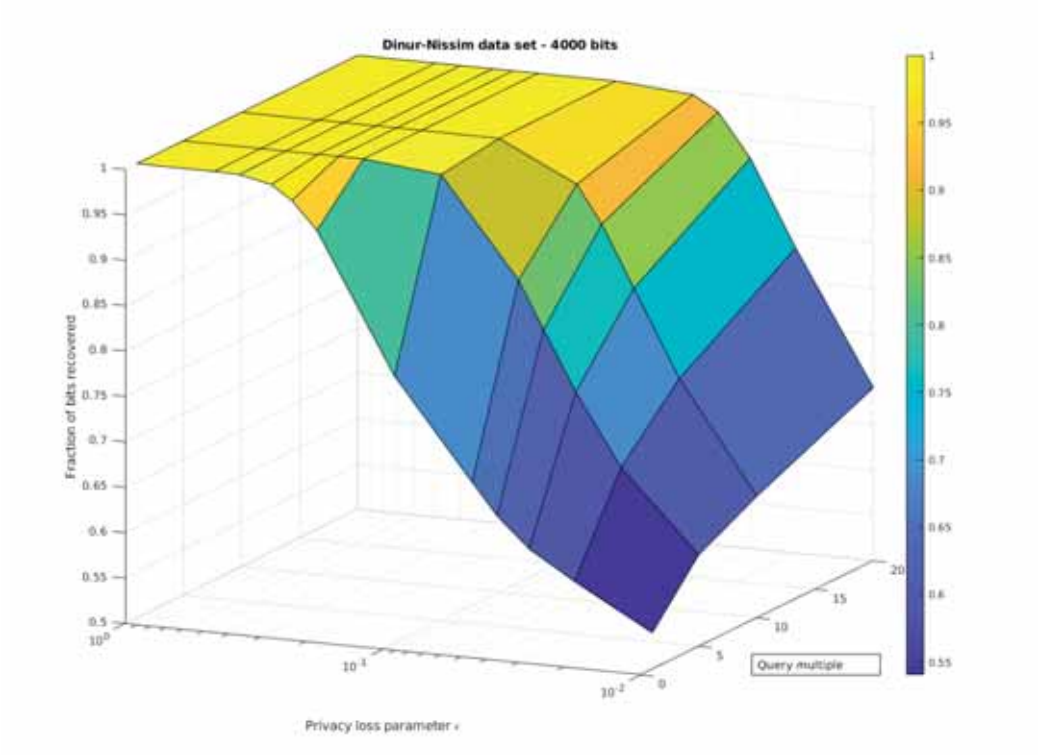


Figure 5-7: Fraction of bits recovered for the Dinur-Nissim database as a function of the privacy loss parameter and the number of multiples of the size of the database.

5.7 Disclosure Avoidance using Differential Privacy

The Census proposes to use an idea similar to that discussed above using the Dinur-Nissim database but applied to the much more complex microdata collected by the Census. As noted above, if one post-processes data that have been previously processed through an algorithm that satisfies the DP conditions, then the post-processed data will also satisfy the constraints of DP provided the original data are not accessed again during the post-processing.

If one creates the usual histograms as published by the Census (i.e. PL94, SF1, etc.) and then applies a DP mechanism to the results, then one could apply the same optimization technique used to demonstrate the Census vulnerability in Section 4 to produce microdata that are now themselves protected by DP. This

approach will create synthetic microdata upon which statistical queries can then be issued. We detail below the proposed approach following closely the briefing to JASON by Dan Kifer [14].

The approach Census will use has three phases

1. Select
2. Measure
3. Reconstruct

The microdata are first represented as a multidimensional histogram H. These are the tables that Census typically publishes. This histogram is then flattened into a column vector. A query on this histogram H is a linear function of the vector and can be represented by a query workload matrix Q. To acquire the answer to a prescribed set of queries we simply compute QH.

Selection phase In the selection phase a strategy matrix A is constructed for the purpose of optimizing the accuracy of various queries. A well chosen strategy matrix will minimize the sensitivity associated with the chosen queries by reducing the statistical variance of the queries. Algorithms for computing such a matrix are given in [20], but require some understanding of what the preferred query workload would be so that the appropriate set of queries is optimized for accuracy.

Measurement phase In this phase the query workload is performed with noise then added to the result. The amount of noise will depend on the sensitivity of the query and the chosen value of ε:

$$\tilde{Y} = AH + \text{Lap}\{\Delta_A/\varepsilon\}$$

where $\tilde{Y}$ is the DP response to the query and Δ_A is a norm measuring the sensitivity of the strategy matrix A.

Reconstruct The final step is to estimate QH from the vector $\tilde{Y}$. This requires undoing the multiplication by the strategy matrix:

$$QH = QA^{+}\tilde{Y}$$

As the strategy matrix may not be square, the Moore-Penrose pesudo-inverse is used to compute H and then QH.

The measurement phase consumes the privacy budget. Once this is accomplished the results could in principle be released to the public. The reconstruction phase will not re-access the private data and hence does not require additional privacy budget. The cleverness of this idea is that the final product can even be in the form of microdata which can then be reprocessed by users of the Census data. What is less clear however, is the accuracy of queries that have not been optimized using the High Dimensional Matrix Method and whether the results of those queries will have an acceptable statistical utility. This will be discussed further in Section 6.

While the steps of this procedure are easily described, the computational aspects of doing this for the census pose significant challenges. Recall that for the country Census publishes billions of queries and so the histogram will have billions of cells. The query matrix could be as large as the square of the histogram size depending on what measurements are to be reported. Choosing a strategy matrix based on the potential query workload is not feasible. The reconstruction is also going to entail an enormous computational cost as a a result of the matrix sizes. Finally, the result of the multiplication by the Moore-Penrose inverse will lead to non-integer results. If we wish to convert these to sensible microdata a second phase will be required in which the results of the first phase will have to be converted to integers. Once this is done the optimization approach taken by Census to reconstruct the microdata can be used to create differentially private microdata.

The solution to the challenges discussed above are to break the problem up into pieces and then perform the DP reconstruction on each piece. The first

attempt to do this was a “Bottom Up” approach in which the select-measure-reconstruct approach was applied to each Census block and then converted to microdata. This has the advantage that the operations are all independent for each block and the privacy budget is simple - one value of ε can be assigned to each block. The privacy cost does not depend on the number of blocks as each of these is processed independently of the others. It also has the advantage that the counts at various levels of the Census hierarchy are consistent. However, the injection of the DP noise adds up as the data are combined to form results for block groups, tracts, etc. A county in a populous region that contains many blocks will have an error proportional to the number of blocks. The “Bottom Up" approach is easy to conceptualize but it doesn’t use the privacy budget efficiently.

Instead, Census will use a “Top-Down” approach. The privacy budget is split into six parts: national, state, county, tract, block group and block. A national histogram $\tilde{H}^0$ is then created using the select measure and reconstruct algorithm outlined above. This involves the population of the US but the number of queries is now manageable as the queries are not specified over geographic levels finer than the nation. Once this protected histogram is in place the same process can then be applied for the states using the privacy budget allocated for states. These histograms are constrained so that they are consistent with national totals. This process is then followed down to the county, block group and finally the block level. Once a protected histogram with non-negative integer entries is created it can then be transformed to microdata using the optimization approach Census used to determine the reconstruction vulnerability as discussed in Section 4. The Top-Down approach has the advantage that it can be performed in parallel and the selection of queries can be optimized at each level making it possible to use the privacy budget more efficiently. It also has the advantage that it enforces any sparsity associated with 0 populations at various levels (for example someone over 100 who indicates they are a member of five racial categories). These are known as structural zeros.

In producing an appropriate histogram that can be turned into microdata two

optimizations are performed. The first is a least squares optimization which effects the Moore-Penrose inverse subject to various constraints that the histogram being determined must be consistent with the parent histogram. For example the total population of the states must sum to the population of the country. The result of this optimization leads to fractional entries and so the second step is to perform an optimization that assigns integer values to the histogram cells such that the entries are non-negative integers that are rounded values of the fractional results and that sum to the same totals consistent with the parent histograms. This "rounding" step is performed using the Gurobi solver [12].

A complication in executing the TopDown algorithm is the need to publish some data without protection. These correspond to the invariants discussed in Section 5.3. Census plans to provide accurate counts of the population of each of the 50 states, DC and Puerto Rico to support apportionment of Congressional representatives. It might also be desirable to report correct population down to the census block.

But in addition, there are other constraints and so it would be desirable to be consistent with these. For example, the number of occupied group quarters and housing units in each census block is public information as a result of a program called Local Update of Census Addresses (LUCA). This program is used by Census to update the Master Address File (MAF) used to distribute census surveys. The addresses themselves are protected under Title 13 but the number of group quarters is publicly released. As a result, if a census block were to have an occupied jail then the TopDown algorithm must assign at least one person to that jail. As another example, the number of householders in a block should be at least the number of households [14]. There are other data-independent constraints. For example, if a household has only one person in it then that person is presumably the householder.

Census has proposed a partial solution to this problem by casting the constraints as a series of network flows that can then be appended to both the least

squares and rounding optimizations described above [14]. This work is still experimental at the time of this writing and will be further evaluated.

The enforcement of invariants such as national and state populations presents no issues in terms of the DP computation. Neither does the enforcement of structural zeroes such as there cannot be any males in a dormitory that is all female. But the constraints that are independent of the data such as the fact that a grandparent must be older than the children in a household creates issues of infeasibility as the optimization recurses down the Census geographic hierarchy. If such implied constraints are ignored there is the possibility that for example assignments at the block group level are not consistent when extended to the higher Census tract level. When this happens it is called a "failed solve"and Census then applies a "failsafe" optimization. The constraints impeding solution are relaxed and the optimizer finds the closest feasible solution meaning a violation of the exact constraint will be allowed. The assignments at the higher geographic level (for example the county level of optimization at the tract level fails) are then modified to maintain hierarchical consistency. The overall impact of the use of the failsafe on the utility of the protected Census data is still not fully understood and is an area of ongoing research. One approach that would avoid this difficulty is to not insist on hierarchical consistency at the finer geographic levels, in particular census blocks. For example providing the correct population in each block might not be enforced as a constraint. This however may have implications for the use of census data in the redistricting process, an issue we discuss in Section 6.

The new disclosure avoidance scheme will now look as in Figure 5-8. It is expected that Census will still perform the usual imputations associated with households and general quarters for which Census enumerators cannot obtain information but, at present, no household swapping will be performed. Instead the Census will apply the TopDown algorithm and then create a set of noised tabular summaries and also, for the first time, the synthetic microdata associated with the summaries.

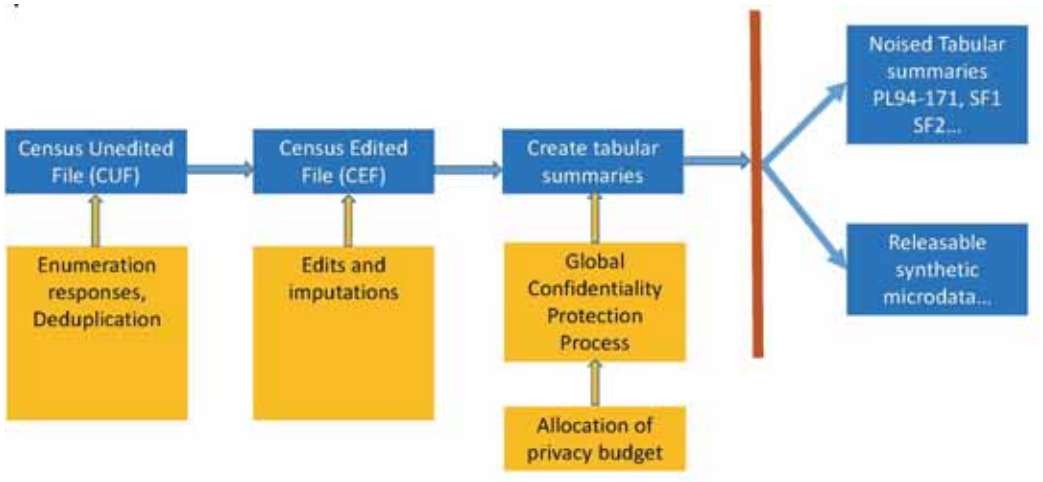


Figure 5-8: A graphical representation of the proposed DAS using the TopDown DP algorithm.

The proposed disclosure avoidance system using DP has been implemented in Python and is publicly available [34]. Work continues to improve query accuracy and enforce invariants and implied constraints. Census is to be commended for making this software available to the community so that it can be examined in detail and inform users on the details of the application of DP to census data.

6 ASSESSING THE ACCURACY-PRIVACY TRADE-OFF

In this section we examine the results of some of the early applications of the new Census DAS on census data. As mentioned in Section 5 Census has publicly released the DAS software. To further aid users, it has processed census data from 1940 and produced synthetic microdata. It has also released some preliminary assessments of query accuracy for the 2010 census data. We discuss these results here with an emphasis on the trade-off between query accuracy and the level of privacy protection.

6.1 Census Analysis of 2010 Census Data

Census has applied the proposed DAS using DP to the 2010 census data. The advantage here is that the schema for the 2010 census largely overlap with the schema for the forthcoming 2020 census. But a disadvantage is that this data is not yet publicly available. By law census data can only be publicly released no earlier than 72 years after a census is taken so the latest data available to the public is the 1940 census. We are able to provide only a limited view of the results of the Census analyses on 2010 data as most of these are not yet available for release and are still protected under Title 13. JASON did have access to these results but the assessment provided here can only describe them qualitatively.

As briefed to JASON by P. LeClerc [16], Census has executed the TopDown algorithm on a histogram from the Census Edited File H_{CEF} to produce a noised histogram of privatized results H_{DAS}. The experiments were performed for the PL94-CVAP product that has 4032 entries representing a shape of $8 \times 2 \times 2 \times 63 \times 2$. Recall that this product is used to examine voting districts to ensure adherence to the Voting Rights Act and includes the following pieces of information:

- 8 group quarters-housing units levels,

- 2 voting age levels,
- 2 Hispanic levels,
- 63 OMB race combinations,
- 2 Citizenship levels.

For each state one can create such a histogram and examine it at various geographic levels: state, county, tract, block group and block. For each geographic level (geolevel) γ, Census executed 25 trials of the DAS, averaged over the results, and reported a number of metrics. We will consider here only one of them:

$$\text{TVD}_\gamma = 1 - \frac{L^1(H_{DAS,\gamma}, H_{CEF,\gamma})}{2\text{POP}_\gamma}.$$

This can be thought of as a type of accuracy metric using the L_1 norm or sum of the magnitudes of the distance between the DAS and CES entries. This is similar in some respects to the Dinur-Nissim query accuracy metric discussed in Section 5.5. If the DAS and CEF histograms were to agree across all components at a given geographic hierarchy level γ, the TVD value would be exactly 1. The possible difference between the values is normalized by twice the population, but this does not provide an absolute lower bound on the TVD metric and it can become negative depending on how much noise is infused into the histogram values.

As of the date of this report, Census has publicly released TVD metrics for the state of New Mexico [30]. These indicate query accuracy vs. privacy loss for actual Census data and may be reflective of the results of the future 2020 Census. In Figure 6-1, the TVD metric as a function of ε is plotted at the state, county, tract group, tract, block group and block for the state population. As ε increases from 0, the TVD metric will tend to one indicating that as ε increases less noise is injected into the histograms until at sufficiently large ε the DAS and CEF results agree in this norm. As can be seen, for geolevels with large populations (e.g. counties, tracts and even block groups) the TVD metric for population is close to one for values of ε as small as 1/2. At even lower levels of ε we see the same

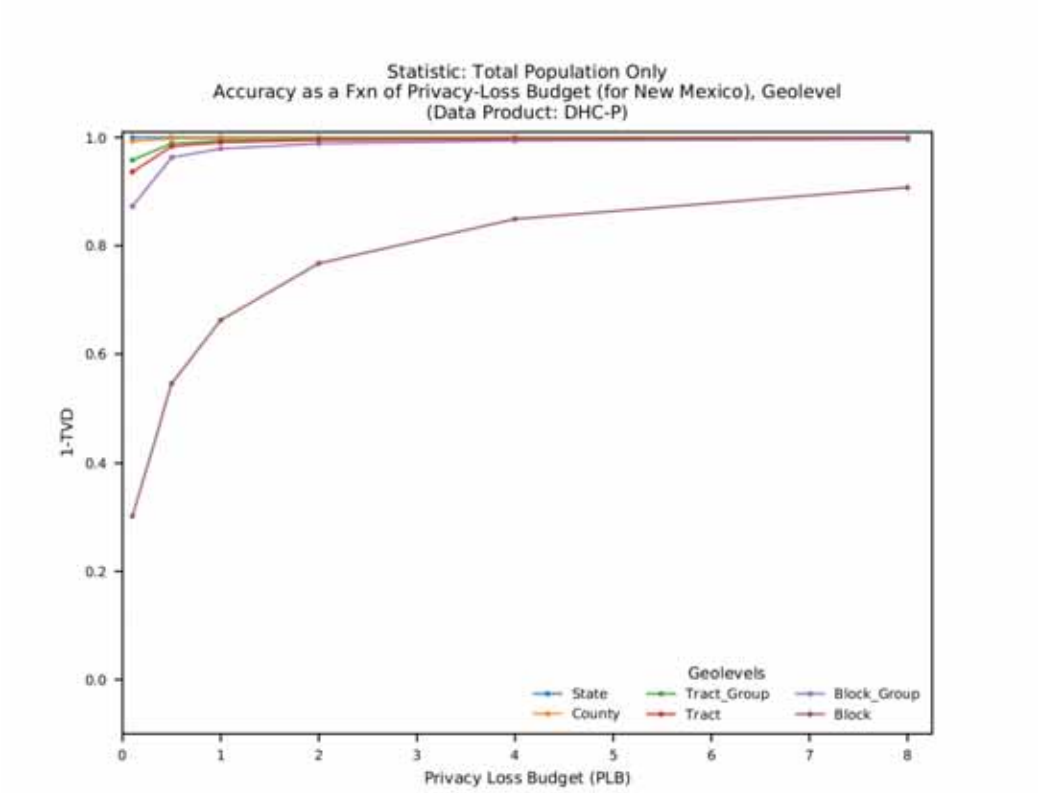


Figure 6-1: A plot of the TVD metric for total population for various geolevels as a function of privacy loss parameter for the state of New Mexico [30].

type of degradation of query accuracy as in the Dinur-Nissim example. Because we cannot tie TVD to a measure of statistical accuracy we cannot comment on whether such degradation of accuracy would or would not be acceptable from that point of view. At the block level, because populations are typically much smaller than block groups the degradation is noticeable and even at $\varepsilon = 4$ we still have TVD ≈ 0.8.

In Figure 6-2 we show again the TVD metric but this time for a subhistogram looking only at those entries associated with race and Hispanic origin. Typically the counts here will be smaller particularly as we examine the finest block level and so the TVD metric deviates further from 1 than shown in Figure 6-1 as the privacy loss budget is decreased.

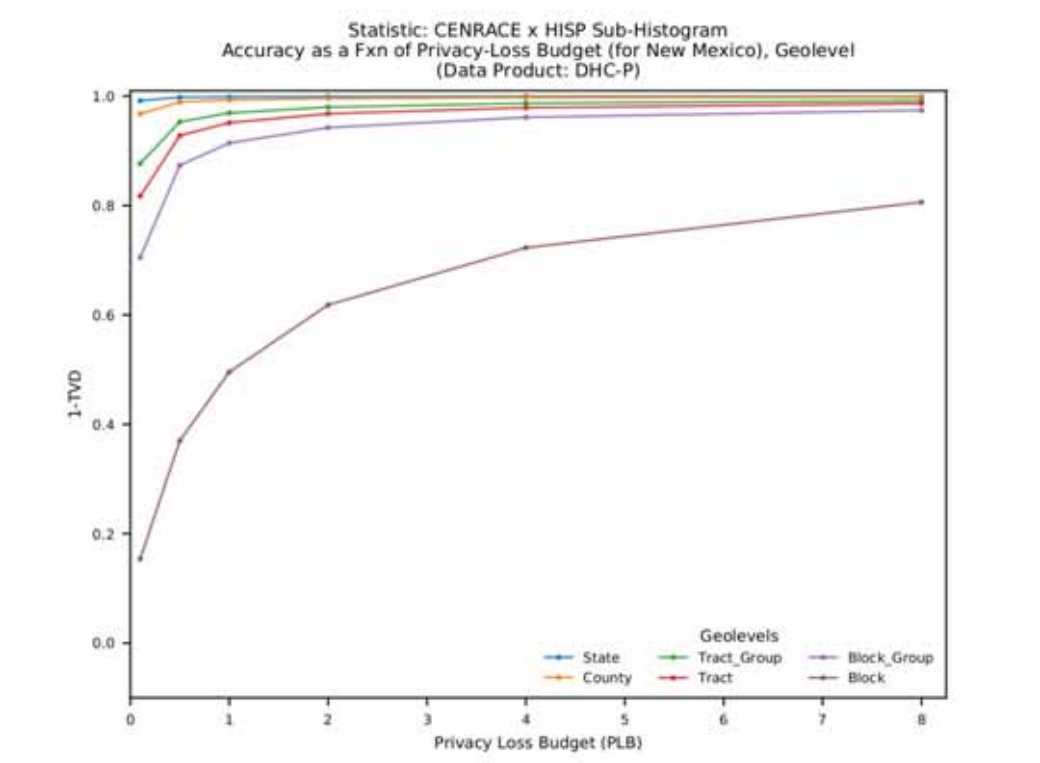


Figure 6-2: A plot of the TVD metric for race and Hispanic origin for various geolevels as a function of privacy loss parameter for the state of New Mexico [30].

The TVD metric provides some insight into the degradation of query accuracy as the privacy loss budget is decreased, but it suffers from being a coarse measure of accuracy as it sums over the entries at a given geolevel and so does not provide a view of the variance of the individual differences. For example, it would be useful to see the distribution of TVD measure block by block. A more detailed assessment in terms of microdata but for the older 1940 Census is discussed in the next section.

6.2 IPUMS Analysis of 1940 Census Data under the Census DAS

IPUMS (Integrated Public Use Microdata Series) is an organization under the University of Minnesota Population Center providing census and survey data from a

variety of countries. It is the world's largest repository of census microdata. JASON was briefed by Dave van Riper of IPUMS [36] (cf. also [37]) who examined in detail the application of the Census DAS to the 1940 Census microdata. We note that JASON has not verified this work but we discuss it here to give examples of the differences between counts associated with the DAS processed synthetic microdata and the true census microdata. As discussed in Section 4, we expect more dispersion as we descend to finer geographic regions. At the time of van Riper's briefing he had performed comparisons for Minnesota census data. Since then, he has also performed analyses for the entire US and it is this data that we discuss here.

It should be noted that the geographical hierarchy for the 1940 census was different than that used today. The finest level of geographic resolution is what was then called an enumeration district. Enumeration districts are roughly comparable to census block groups on the geographic spine and also similar in some ways to what Census terms "places". The median population for enumeration districts was about 1000 people. The median population for census places in 1940 was about 800 people.

As indicated in Section 5, Census has publicly released differentially private microdata for the 1940 census. Microdata files were generated for the entire country for eight different values of the privacy loss parameter ε : 0.25, 0.5, 0.75, 1.0, 2.0 4.0, 6.0, 8.0. Four runs of the DAS were provide at each value of ε. The microdata made available are those of the PL94-CVAP Census product and include whether a respondent is of voting age, Hispanic origin and Race as well as household and group quarters type at four geographic levels: national, state, county and enumeration district. IPUMS did not run the Census DAS to generate synthetic microdata. Instead it analyzed those results generated by Census to compare against unfiltered microdata that constitute ground truth. The source code for the DAS system [34] is configurable so that one can allocate fractions of the total privacy budget over the various geographic levels and tables. In this case the budget is allocated evenly over geographic levels. Each level of the hierar-

chy receives a quarter of the total privacy budget. Allocations must also be made for the various tables that are produced and then subsequently noised by the DP algorithm. In this case Census chose the following fractions:

- Voting age by Hispanic Origin by Race: 0.675
- Household group quarters type: 0.225
- Full cross of all variables: 0.1

The fraction of the total privacy budget to be allocated for each level and for each table is then the product of the geolevel allocation times the table fractions. For a given total privacy loss budget ε it is these fractions that are used to provide the noise levels for each individual table at a given geographic level. For example if the total privacy budget were 0.25 then the privacy budget for each histogram will look as shown in Table 6-3. The table shows the effective values of ε but also the level of dispersion for an equivalent Laplace distribution. These dispersion levels will affect various tables differently. A table associated with large counts will not be significantly affected by an ε corresponding to a dispersion of 300 but a table at the enumeration district level could be significantly affected.

Box plots of the distribution of populations across all US counties in 1940 are shown in Figure 6-3 for all the values of ε used in the Census runs of the DAS. The distribution as computed by IPUMS from the true 1940 microdata is shown at the left of the Figure. As can be seen, as ε increases the box plots converge to the IPUMS result. For the lowest value of ε used, differences can be seen for populations of 100 or more. By and large, the box plots are quite similar across the various values of ε. More insight into the effect of the DAS at the finer geolevels can be seen in Figure 6-4 where box plots for the differences between the DAS and IPUMS population estimates are shown. The orange box plots represent counties and the teal plots represent enumeration districts. Again as ε increases we see the differences reduce. But at lower values of ε differences on the order of several hundred people appear when we look at various outliers. It should be noted

Geography level	Table	ϵ for Table at level	Noise dispersion
Nation	Vot–Hisp-Race	0.042	47.4
Nation	HouseholdGenQuart	0.014	142
Nation	Detailed	0.006	320
State	Vot –Hisp-Race	0.042	47.4
State	HouseholdGenQuart	0.014	142
State	Detailed	0.006	320
County	Vot –Hisp-Race	0.042	47.4
County	HouseholdGenQuart	0.014	142
County	Detailed	0.006	320
Enum Dist	Vot–Hisp-Race	0.042	47.4
Enum Dist	HouseholdGenQuart	0.014	142
Enum Dist	Detailed	0.006	320

Table 6-3: Values of the privacy budget allocated to the various geolevels and tables by the Census DAS system for the 1940 Census data [36]. The noise dispersion is listed here to give some notion of the variance of the noise applied to the data. In this case the value $\varepsilon = 0.25$ is used [36]

.

that the box plots are not normalized and that the teal box plots for enumeration districts are smaller simply by virtue of representing smaller populations.

Van Riper has also computed how the populations of counties compare in detail in Figure 6-5. The Figure plots the IPUMS value for a county population vs. the DAS value. The level of agreement is measured by how closely the two values would lie to the 45° line indicating equality. As can be seen the county populations align well at all values of ε. In contrast, for enumeration districts we see in Figure 6-6 more dispersion. This is most observable as ε becomes smaller. Note that because the DAS does not allow negative population there is a pile-up as population size decreases. Such results are to be expected as one focuses on finer geolevels and smaller populations.

The same analysis has been performed for population under 18 across all US counties for the 1940 Census. These are shown in Figure 6-7. This too looks quite

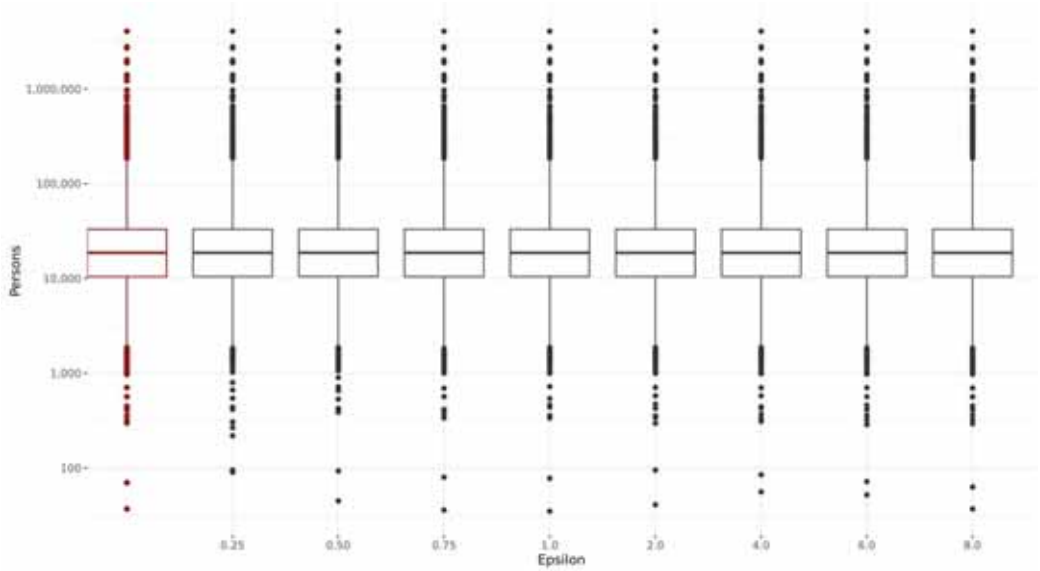


Figure 6-3: Box plots for the distribution of total US population in 1940 under different values of the privacy loss parameter [36].

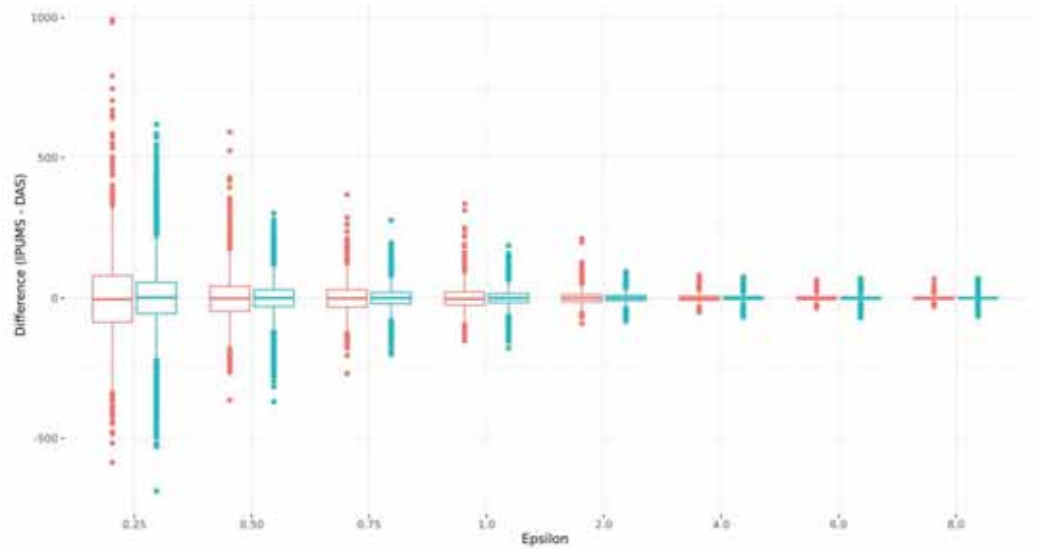


Figure 6-4: Box plots for the differences between IPUMS and Census DAS for total population counts under different values of the privacy loss parameter [36].

similar to population estimates with some issues seen for counties with smaller populations at lower values of ε. The corresponding results for enumeration districts are shown in Figure 6-8. Because we are now focusing on a subgroup of the population for enumeration districts there is yet more dispersion in the results. But

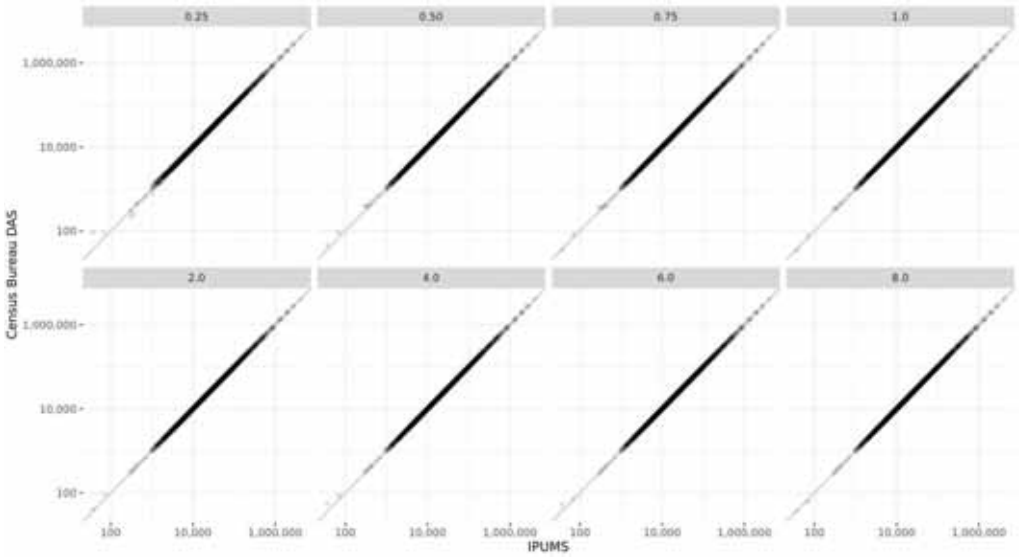


Figure 6-5: Total population for US counties under differing levels of the privacy loss parameter [36].

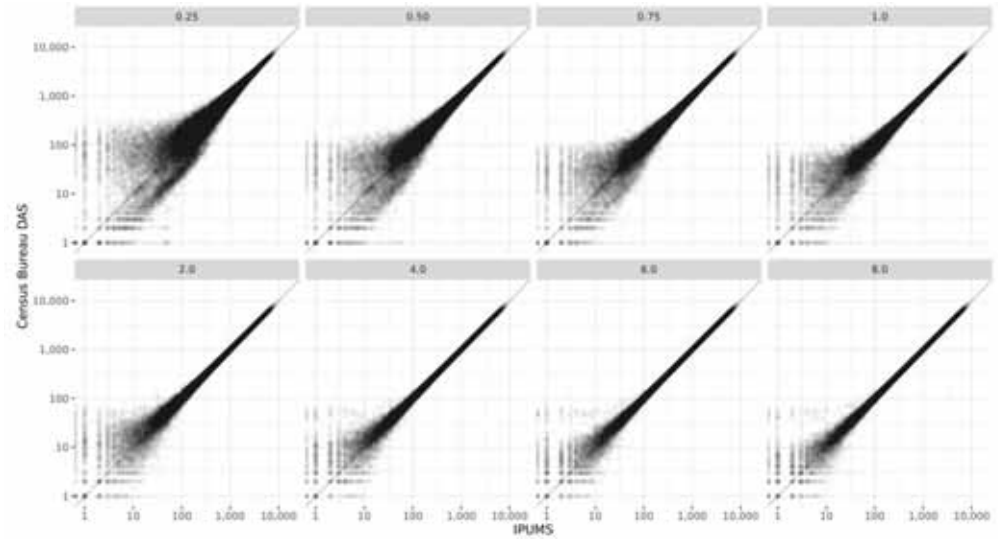


Figure 6-6: Total population for US enumeration districts under differing levels of the privacy loss parameter [36].

perhaps of some concern is that in some enumeration districts the DAS indicates a large number of people under 18 when there are in fact very few. There are some enumeration districts with 50 or more people where this particular application of the DAS (with values of ε of 0.25, 0.5 and even in some cases 1.0) indicates that 100% of the population is under 18, an observation that could have implications

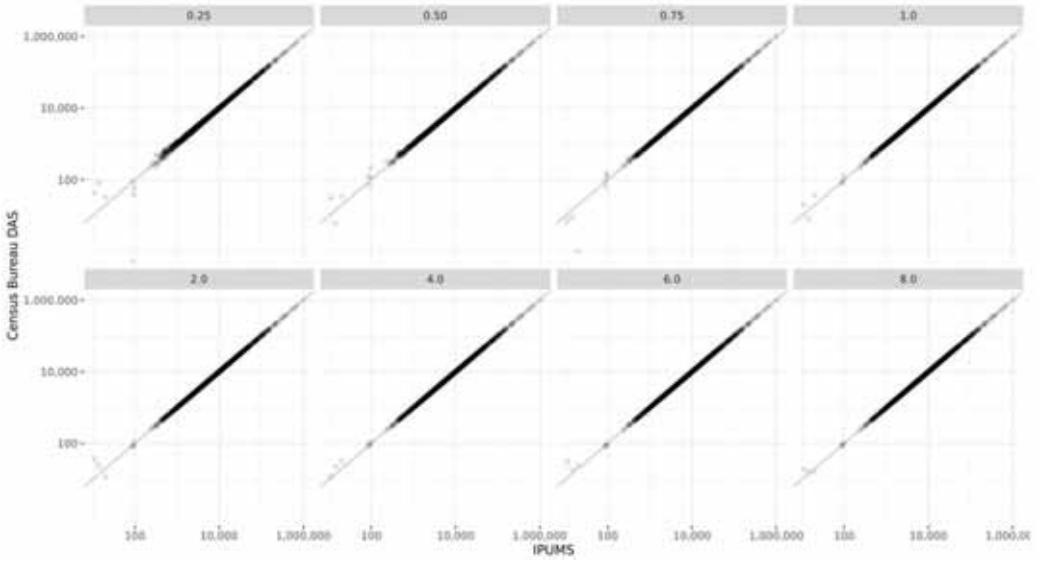


Figure 6-7: Total population under 18 for US counties under differing levels of the privacy loss parameter [36]

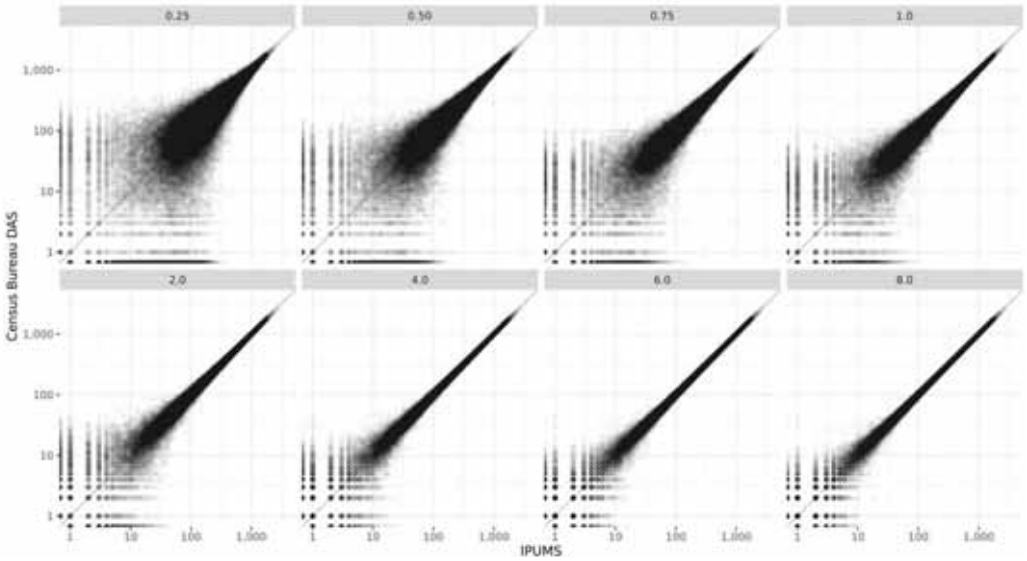


Figure 6-8: Total population under 18 for enumeration districts under differing levels of the privacy loss parameter [36]

for assessments of voting age population, a component of the information needed for the PL94 publication.

Several points should be emphasized in examining the current application of the DAS:

- The DAS does not unduly perturb statistics at the national, state and even largely at the county level at all the values of ε considered.

- The dispersion seen in the IPUMS-DAS comparison for enumeration districts is to be expected at lower values of ε. The DAS is after all meant to protect small populations.

- The application of the DAS will degrade the utility of various statistics. This degradation will increase as one further restricts the population by characteristics such as race, voting age, etc. This illustrates a trade-off inherent in the use of DP among privacy, accuracy and granularity of queries. The requirements for accuracy will need to be determined in the future through consultation with external users of the data. We discuss this trade-off further in Section 7.

- The allocation of the privacy budget can be modified depending on the accuracy requirements. For example it would be possible to allow for larger privacy loss parameters for some tables and less for others provided the total privacy budget is conserved.

- The current version of the DAS is a demonstration product. For example, at the time of this writing, the implementation presented here does not benefit from the improved accuracy of the high dimensional matrix method. Nor do the products contain all the invariants and constraints that the Census bureau has identified. Work is in progress to improve query accuracy to the extent possible. As these improvements are made it will be important to continue to reevaluate the performance of the DAS against ground truth.

7 MANAGING THE TRADE-OFF OF ACCURACY, GRANULARITY AND PRIVACY

Published census tabulations must balance inconsistent desiderata. They should be accurate (i.e., published counts should be the sums of the underlying micro-data). But tabulations should also be appropriately granular (i.e., have a high level of detail such as block, gender, age, race/ethnicity, etc. But, as has been discussed, pushing granularity to the extreme can create small (or even singleton) counts in table entries (particularly in small blocks), thereby eroding privacy. Of course, privacy could be enhanced and granularity preserved by relaxing the accuracy requirement (as embodied in DP or swapping schemes). Alternatively, privacy could be enhanced and accuracy preserved by reducing granularity. The situation can be illustrated by the "disclosure triangle", where the balance among the three competing considerations of privacy, accuracy, and granularity varies across the interior as shown in Figure 7-1.

No compromise will be perfect. In this section, we discuss some aspects of managing this trade-off.

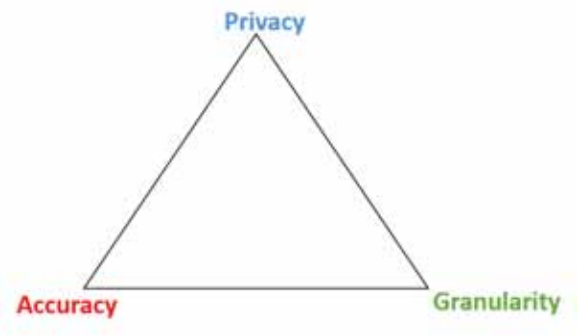


Figure 7-1: Census must balance, accuracy, granularity and privacy in its publications. It is not possible to achieve all three simultaneously.

7.1 Risk Assessment

The use of DP is clearly promising as a way to protect census data, but it is important to recall the original motivation for its use. Its proposed use was primarily motivated by the 17% re-identification rate assessed by Census using the 2010 tables, and thus the degree to which DP prevents re-identification needs to be similarly explored. Technically, differential privacy as pointed out by Reiter [28] is a guarantee

> "on the incremental disclosure risks of participating (in a survey) over whatever disclosure risks the data subjects face even if they do not participate (in the survey)".

It does not provide an assessment of disclosure risk in and of itself. It is also not one methodology. A number of algorithms can be applied and must be implemented correctly. In the case of its use for the census there are clearly complications like invariants, implied constraints etc. that will require further work and assessment. For these reasons, explicit quantification of the risk of re-identification is still required. The choice of ε should be informed by calculations of the risk of re-identification using the methods developed by Census and linking with current commercially-available data but applied to microdata as processed through DP. JASON understands that this will be significantly more difficult than the original analysis that led to the re-identification of the 2010 Census data vulnerability. This is because the matching of the microdata in the absence of noise to commercial data was aided by the availability of the geographic location. The synthetic data generated by DP algorithms will not have this feature and so matching to commercial data bases will have to be performed using probabilistic record linkage (cf. for example [9]). A very useful property of DP here is that such linkage can be attempted at various values of ε. At very high values of ε we expect to recover the noise-free values and so we would also verify the previously assessed re-identification level of 17% against commercial marketing databases. But as ε

is decreased this re-identification rate must degrade. An open question is at what value of ε would it degrade to a value sufficiently low so as to be administratively acceptable? While no official value of such a lower bound has ever been provided (nor would we expect one to be) presentations from Census have indicated that the re-identification rate of 17% was viewed as something like four orders of magnitude higher than previously assessed [27].

The fact that methods of data science will improve and commercially available data will become more comprehensive over time does not obviate the need for an analysis that can inform the current decision. Knowing the outcomes based on current data can help to support a choice of ε. Once some assessment of an appropriate "upper bound" for ε based on disclosure risk is in hand, further considerations regarding statistical accuracy for future queries on the data can be made in ultimately deciding the level of noise to be applied to the 2020 data.

7.2 Engaging the User Community

Analyses of aggregate data involving large populations will be minimally impacted by DP. Impacts will increase as one focuses on finer levels of geography or other demographic measures. We emphasize that this is precisely the desired impact of DP because individuals within a smaller group will be more identifiable, and thus it is precisely this "blurring" from DP that protects the privacy of these individuals. This aspect of DP needs to be effectively communicated to future users of Census data.

The challenge is to better quantify the balance of privacy protection and data utility for smaller groups. There are multiple communities with a deep interest in the accuracy-privacy-granularity tradeoff:

State governments and redistricting commissions These bodies are responsible for the drawing of Congressional and State legislative districts. PL94-171 requires the Census to provide to these bodies an opportunity to identify

the geographic areas relevant to redistricting and to then deliver tabulations of the population as well as race, race for population 18 and over (voting age), Hispanicity and Hispanicity for those 18 and over, occupancy status and, in 2020, group quarters population by group quarters type.

Local governments Local governments use census data for redistricting as well as to inform assessments of public health, safety, and emergency preparedness for the residents.

Residents Residents use census data to support community initiatives and to decide where to live, learn, work and play.

Social scientists and economists Census data forms a foundation for demographic studies as well as economic research.

Census has to some extent reached out to these communities through a July 2018 Federal Register Notice as well as several academic conferences [23]. The feedback received by Census emphasized several aspects:

- There was little understanding as to the need for application of Differential Privacy
- Users were vocal about the need to maintain block level data so that custom geographies could be constructed.
- Concerns were voiced about the potential loss of information for small geographic areas.

Clearly more work is needed and Census should participate actively in various fora, working with the community to characterize the scales and types of queries that will and will not be substantially impacted at different values of ε. For example, opportunities for stakeholders to assess accuracy of queries on 2010 census data made available at various levels of protection would go a long way towards helping users assess the impact of DP on future analyses. In general it

will be necessary to engage and educate the various communities of stakeholders so that they can fully understand the implications (and the need for) DP. These engagements should be two-way conversations so that the Census Bureau can understand the breadth of requirements for census data, and stakeholders can in turn more fully appreciate the need for confidentiality protection in the present era of "big data", and perhaps also be reassured that their statistical needs can still be met.

7.3 Possible Impacts on Redistricting

As indicated above, redistricting bodies will require population and other data for regions with populations infused with noise from the DP process. There is concern that the population estimates derived from differentially protected Census block data will lead to uncertainties in designing state and Congressional voting districts. Census has begun to consider these issues, for example, in their recent end-to-end test for the state of Rhode Island [40]. We cannot discuss the variance of the actual counts and those treated under DP quantitatively here as these data are protected under Title 13. But, especially for the counts associated with smaller state legislature districts, the variances may lead to concerns in verifying that the districts are properly sized relative to the requirements of the Voting Rights Act. JASON was briefed by Justin Levitt [18] that such district equalization is a "legal fiction" since it is impossible to guarantee the accuracy and precision of the counts; they are a snapshot in time and so are not temporally static. Overall, the noise from block-level estimates is not expected to lead to legal jeopardy, but could in the case where, for example, racial makeup nears thresholds that elicit concern. Census is currently engaged with the Department of Justice regarding this issue but at the time of the writing of this report, Census has not allayed the Department of Justice's concerns regarding this issue.

7.4 Limiting Release of Small Scale Data

The trade-off between probability of re-identification and statistical accuracy is reflected in the choice of the DP privacy-loss parameter. A low value increases the level of injected noise (and thus also decreases probability of re-identification) but degrades statistical calculations. Another factor that also influences the choice of privacy-loss parameter is the number and geographical resolution of the tables released, an aspect of granularity of the allowed queries. For example, if no block-level data were publicly released, a re-identification "attack" of the sort described above presumably would become more difficult, perhaps making it feasible to add less noise and so allowing a larger value of ε.

For those public officials and researchers needing access to the finer scale block level data, special channels in the form of protected enclaves may be required. We discuss this next in Section 7.5. This most likely cannot be a solution for certain uses of Census data mandated by law. For example, redistricting must be performed in a way that is transparent to the public. Today this requires using block level populations in designing the new districts. These will be infused with noise under differential privacy. While it is thought that these population estimates can still be used for redistricting, their overall utility is closely tied to the value of ε that is ultimately chosen. Too low a value of ε may lead to concern over the totals. This seems to be a particularly difficult problem that must be solved in close consultation with the relevant stakeholders.

7.5 The Need for Special Channels

Depending on the ultimate level of privacy protection that is applied for the 2020 census, some stakeholders may need access to more accurate data. A benefit of DP is that products can be generated at various levels of protection depending on the level of statistical accuracy required. The privacy-loss parameter can be viewed as a type of knob by which higher settings lead to less protection but more

accuracy. However, products publicly released with too low a level of protection will again raise the risk of re-identification.

One approach might be to use technology (e.g. virtual machines, secure computation platforms etc.) to create protected data enclaves that allow access to trusted stakeholders of census data at lower levels of privacy protection. Inappropriate disclosure of such data could still be legally enjoined via the use of binding non-disclosure agreements such as those currently in Title 13. This idea is similar to the concept of "need to know" used in environments handling classified information. In some cases there may emerge a need to communicate to various trusted parties census data either with no infused noise or perhaps less infused noise than applied for the public release of the 2020 census. Examples include the need to obtain accurate statistics associated with state or local government initiatives, or to perform socio-economic research associated with small populations.

At present, the only way to obtain data not infused with noise is to apply for access via a Federal Statistical Research Data Center. These centers are partnerships between federal statistical agencies like the Census and various research institutions. The facilities provide secure access to microdata for the purposes of statistical research. As of January 2018, there were 294 approved active projects with Census accounting for over half of these. All researchers must at present obtain Census Special Sworn Status (to uphold Title 13), pass a background check and develop a proposal in collaboration with a Census researcher.

The use of DP presents an opportunity to expand the number of people who may access more finely-grained data but who would not need to access the original microdata. Products could be constructed at higher levels of the privacy loss parameter than that used in releasing Census data to the public. In a sense, the use of DP allows Census to control the level of detail available to a researcher but in accord with the users "need to know", or more appropriately their need to access data at a given level of fidelity.

If such a program is developed there may arise the need to increase the ca-

pacity of the research data centers but at the same time the requisite security must be enforced. The defense and intelligence communities are facing similar issues and have responded by using cloud-based infrastructure and “thin client" terminals with limited input/output capability and strongly encrypted communication to ensure that data is appropriately protected and not handled improperly.

Transformative work in various areas of social science and economics has resulted from the ability to access and analyze detailed Census data. For example, Chetty and his colleagues [3] have used detailed census data to research approaches to using DP in small areas while maintaining the guarantees of DP. The development of virtual enclaves would expand opportunities to make similar contributions to a much wider cohort of researchers.

8 Conclusion

We conclude this report with a discussion of the controversy that has arisen as a result of the discovery of the Census vulnerability. The need to address the Census vulnerability also brings forward aspects of a tension between laws that protect privacy as opposed to those that require the government to report accurate statistics. We close with a set of findings and recommendations.

8.1 The Census Vulnerability Raises Real Privacy Issues

In the view of JASON, Census has convincingly demonstrated the existence of a vulnerability that census respondents can be re-identified through the process of reconstructing microdata from the decennial census tabular data and linking that data to databases containing similar information that can identify the respondent. The re-identification relied on matching Census records with commercial marketing datasets. These data providers, such as Experian, ConsumerView, and others already have a good deal of the data Census must secure such as name, age, gender, address, number in household, as well as credit histories, auto ownership, purchasing, consumer tastes, political attitudes, etc. But we note that the accuracy and granularity of their data is almost surely less than Census, and they generally do not include race or Hispanic identity; the latter is most likely a choice, not a fundamental constraint on information collection. In addition to this data there is also proprietary data maintained by Facebook, the location data collected by cell phone providers, etc.

One might argue that Census data is not of much additional utility given the limited amount of information gathered in the decennial census. However, many components of the data Census collects are not in the public domain and are still viewed as private information. For example information on children is hard to purchase commercially because its collection is enjoined by laws such as the Children's Online Privacy Protection Act. Other examples include race, number

and ages of children, sexuality of household members and, in the near future, citizenship status. Census has an obligation to protect this information under Title 13 and, in view of the demonstrated vulnerability, it is clear that the usual approaches to disclosure avoidance such as swapping, top and bottom coding, etc. are inadequate. The proposal to use Differential Privacy to protect personal data is promising although further work is requried as this report points out.

The decision to use Differential Privacy has elicited concerns from demographers and social scientists. Ruggles has argued, for example, that Census has not demonstrated that the vulnerability it discovered is as serious as claimed. In [29] he states

> "In the end only 50% of the reconstructed cases accurately matched a case from the HDF source data. In the great majority of the mismatched cases, the errors results from a discrepancy in age. Given the 50% error rate, it is not justifiable to describe the microdata as 'accurately reconstructed'."
>
> Reconstructing microdata from tabular data does not by itself allow identification of respondents allow identification of respondents; to determine who the individuals actually are, one would then have to match their characteristics to an external identified database (including, for example, names or Social Security numbers) in a conventional re-identification attack. The Census Bureau attempted to do this but only a small fraction of re-identifications actually turned out to be correct, and Abowd ... concluded that 'the risk of re-identification is small.' Therefore, the system worked as designed: because of the combination of swapping, imputation and editing, reporting error in the census, error in the identified credit agency file, and errors introduced in the microdata reconstruction, there is sufficient uncertainty in the data to make positive identification by an outsider impossible."

This statement may reflect the state of affairs prior to the re-identification ef-

fort of the Census discussed in Section 4.1 that succeeded in re-identifying 17% of the US population in 2010. An earlier re-identification attempt by the Census had some issues matching the Census geo-ids with those of commercial data. Once this was understood and fixed, the results discussed in Section 4.1 were obtained.

Ruggles also argues that use of differential privacy will mask respondents characteristics, data that are valuable in demographic and other studies. He correctly asserts that masking characteristics is not explicitly required under the law. But Census is prohibited from publishing

> "any representation of information that permits the identity of the respondent to whom the information applies to be reasonably inferred by either direct or indirect means..."

Given the level of re-identification that was achieved in the Census vulnerability study, it is certainly arguable that releasing tabular information without noise such that the microdata can be reconstructed and possibly matched with external data makes the tabular information just such a representation.

Ruggles further argues that Census would not validate any potential re-identification. This is true, but the fact remains that a commercial data provider can still perform the re-identification attack, then perform a probabilistic record match (perhaps using data held out from the re-identification), and, if the result looks sufficiently promising, add this to their database along with extra information on race, children, sexuality, etc. The argument that Census will not confirm the re-identification is true whether one performs any disclosure avoidance or not. But it is still the responsibility of Census not to abet such re-identification. Finally, there is the issue of whether Census data (as opposed to ACS data) is particularly sensitive. It can be argued that knowledge of various characteristics combined with location data could certainly be abused in various instances and so this provides further support that Census should enforce privacy of census data.

Even more concern has been voiced in the social science and demographer

communities regarding the possibility that the ACS tables and microdata sample may also now require similar protection. To date Census has not established that a similar vulnerability exists for the ACS data. Intuitively, it *should* be harder to re-identify this data as it is a small sample of the population and what is released is carefully chosen so as to preserve confidentiality. In any case, no plan by Census exists at present to apply methods of formal privacy to the ACS, and no changes are envisioned in the format for data release at least until 2025 when the issue will be reconsidered (cf. for example, [33]).

8.2 Two Statutory Requirements are in Tension in Title 13

It is to be expected that advances in technology may introduce tensions or conflicts among statutory provisions that were seen as conflict-free when they were enacted in the past. Under the Executive Branch's broad powers to interpret and apply the law, responsibility falls on Executive agency government officials to set policies that attempt to "square the circle" in a defensible manner, even when no perfect solution is possible. Such policies, both as to the procedure of how they are set and their substance, are potentially subject to judicial review, e.g., under the Administrative Procedures Act (5 USC Section 500). The resolution of statutory conflicts is thus ultimately a matter for the courts, or for Congress if it chooses to change the law.

In the above light, we examine two statutory provisions of Title 13. Section 214 ("Wrongful disclosure of information") provides

> "[No official] may make any publication whereby the data furnished by any particular establishment or individual under this title can be identified..."

There is little or no case law to guide us in the interpretation of what, at first sight, seems a clear provision. But how clear is it? Does "whereby" mean by itself

without reference to other sources of (e.g., commercial) data? Or does "whereby" mean may not add, even incrementally in the smallest degree, to the likelihood that an individual can be identified using commercially available data? Or is it something in-between? What about "can be identified"? Does this mean identified with certainty? Or does it mean identified probabilistically as more likely than other individuals? And, if the latter, what is the quantitative level of probability that is prohibited?

Census has traditionally adopted very strict interpretations of Section 214 for a host of good reasons, including that doing so encourages trust and participation in the census. Section 141 (Public Law PL 94-171) specifies a process by which the states propose, and the Secretary of Commerce agrees to, a geographical specification of voting districts within each state[3]. It then requires that

> "Tabulations of population for the areas identified in any plan approved by the Secretary shall be completed by him as expeditiously as possible after the decennial census date and reported to the Governor of the State involved and to the officers or public bodies having responsibility for legislative apportionment or districting of such State ... "

The plain-language meaning of "tabulation of population" is fairly obvious: one counts the number of persons satisfying some required condition(s) and enters that number into a table. At the time of the 2010 Census, and with the disclosure avoidance procedures adopted at that time, there seemed to be no significant conflict between the statutory requirements of Section 214 and Section 141. Swapping, for example, preserves population counts in any geographical area. To the extent that swapped individuals were matched for other characteristics (e.g., voting age), counts of persons with matched characteristics would also be preserved. Finally, the use of swapping may allow for the use of a larger value of ε used for

[3]Technically the law says "...the geographic areas for which specific tabulations of population are desired". This has been identified as blocks and voting districts since the law was passed

publication of the various tabulations. This would have to be determined through an empirical assessment of re-identification risk performed both with and without swapping.

Census has determined, and JASON agrees, that swapping alone is an insufficient disclosure avoidance methodology for the 2020 Census. The proposed use of DP in the 2020 Census, which is by now almost certain, will bring the mandates of Section 214 and Section 141 into conflict to a substantially greater degree than previously. Although Census proposes to impose invariants along a backbone of nested geographical regions, the revised state voting districts mayh not be on this backbone, and hence will be subject to count errors whose magnitude depends on the amount of DP imposed (i.e., the choice of ε).

There is no perfect resolution of the conflict. JASON heard the opinion of some experts outside of government that inaccuracies as large as 1000 persons in state voting district counts are acceptable. However, we also heard that, in many cases, the actions of state officials can be interpreted as indicating a mistaken belief that the counts are much more accurate than this. We are not aware of any case law or judicial guidance on the issue. Thus, Census will need to adopt a policy that is a sensible compromise between conflicting provisions of law, recognizing that the ultimate adjudication of such a policy - should it prove to be controversial - lies elsewhere. Too small a value of ε, while more perfectly satisfying Section 214, satisfies Section 141 less perfectly, both being statutory requirements.

We conclude this report with JASON's findings and recommendations.

8.3 Findings

8.3.1 The re-identification vulnerability

- The Census has demonstrated the re-identification of individuals using the published 2010 census tables.

- Approaches to disclosure avoidance such as swapping and top and bottom coding applied at the level used in the 2010 census are insufficient to prevent re-identification given the ability to perform database reconstruction and the availability of external data.

8.3.2 The use of Differential Privacy

- The proposed use by Census of Differential Privacy to prevent re-identification is promising, but there is as yet no clear picture of how much noise is required to adequately protect census respondents. The appropriate risk assessments have not been performed.

- The Census has not fully identified or prioritized the queries that will be optimized for accuracy under Differential Privacy.

- At some proposed levels of confidentiality protection, and especially for small populations, census block-level data become noisy and lose statistical utility.

- Currently, Differential Privacy implementations do not provide uncertainty estimates for census queries.

  As has been seen in Section 6, as the geographic resolution becomes finer, DP will by design affect query results. In such cases, there will at least be a need to inform users of the variances associated with a given query. While the amount of noise injected into tables is known as a result of the open publication of the privacy budgets, the variance in a query is also affected by the size of the population involved in answering that query, the use of the high-dimensional matrix method, the enforcement of invariants, etc. complicating the error analysis. Error assessment could be accomplished by performing multiple instances of a query and then assessing the variation of the results, but this requires re-accessing the data and so potentially violating the DP bounds. Ashmeade [2] has proposed an approach to

estimate query error by using the post-processed results and then assessing variance using those results. This has the advantage that one need not access the confidential data. Ashmeade presents some empirical evidence that, for the most part, this approach yields sensible bounds, but for small privacy budgets occasional outliers occur and the results of such an estimate vary widely from the true results obtained using Monte-Carlo methods. This issue clearly requires further work.

8.3.3 Stakeholder response

- Census has not adequately engaged their stakeholder communities regarding the implications of Differential Privacy for confidentiality protection and statistical utility.
- Release of block-level data aggravates the tension between confidentiality protection and data utility.
- Regarding statistical utility, because the use of Differential Privacy is new and state-of-the-art, it is not yet clear to the community of external stakeholders what the overall impact will be.

8.3.4 The pace of introduction of Differential Privacy

- The use of Differential Privacy may bring into conflict two statutory responsibilities of Census, namely reporting of voting district populations and prevention of re-identification.
- The public, and many specialized constituencies, expect from government a measured pace of change, allowing them to adjust to change without excessive dislocation.

8.4 Recommendations

8.4.1 The re-identification vulnerability

- Use substantially equivalent methodologies as employed on the 2010 census data coupled with probabilistic record linkage to assess re-identification risk as a function of the privacy-loss parameter.
- Evaluate the trade-offs between re-identification risk and data utility arising from publishing fewer tables (e.g. none at the block-level) but at larger values of the privacy-loss parameter.

8.4.2 Communication with external stakeholders

- Develop and circulate a list of frequently asked questions for the various stakeholder communities.
- Organize a set of workshops wherein users of census data can work with differentially private 2010 census data at various levels of confidentiality protection. Ensure all user communities are represented.
- Develop a set of 2010 tabulations and microdata at differing values of the privacy-loss parameter and make those available to stakeholders so that they can perform relevant queries to assess utility and also provide input into the query optimization process.
- Develop effective communication for groups of stakeholders regarding the impact of Differential Privacy on their uses for census data.
- Develop and provide to users error estimates for queries on data filtered through Differential Privacy.

8.4.3 Deployment of Differential Privacy for the 2020 census and beyond

- In addition to the use of Differential Privacy, at whatever level of confidentiality protection is ultimately chosen, apply swapping as performed for the 2010 census so that no unexpected weakness of Differential Privacy as applied can result in a 2020 census with less protection that 2010.

  There is always the possibility that unforeseen issues or implementation errors may lead to violations of the privacy protections that DP aims to enforce. Such things have happened in the past, for example, in the cryptographic community. JASON recommends that Census apply the traditional disclosure avoidance procedures applied in the 2010 census and then apply DP on top of this dataset. The advantage in JASON's view is that one can communicate that DP is a proposed improvement over traditional approaches and, should there arise any issue with DP, the previously used protections will still be in force. The software infrastructure for the traditional disclosure avoidance approach would have to be reconstructed and this could prove to be a challenge.

- Defer the choice of the privacy-loss parameter and allocation of the detailed privacy budget for the 2020 census until the re-identification risk is assessed and the impact on external users is understood.

- Develop an approach, using real or virtual data enclaves, to facilitate access by trusted users of census data with a larger privacy-loss budget than those released publicly.

- Forgo any public release of block-level data and reallocate that part of the privacy-loss budget to higher geographic levels.

- Amid increasing demands for more granular data and in the face of conflicting statutory requirements, seek clarity on legal obligations for protection of data.

A APPENDIX: Information Theory and Database Uniqueness

> Je n'ai fait celle-ci plus longue que parce que je n'ai pas eu le loisir de la faire plus courte.
> *(I'd not have made this [letter] so long, had I had time to make it shorter.)*
>
> Blaise Pascal, *Lettres Provinciales*, 4 Dec. 1656.

In this appendix we examine the Dinur-Nissim (DN) results in the context of information theory. As a reminder, DN idealize a database as a string $d = (d_1, \ldots, d_n)$ of n bits, and a *noiseless* query as the sum of a specified subset of those bits; that is to say, the answer to the query is

$$A(q) = \sum_{i \in q} d_i \equiv \boldsymbol{w}_q^T \boldsymbol{d} \tag{A-1}$$

In the second form above, the string d is represented by a column vector $\boldsymbol{d}$, whose components are either 0 or 1, while $\boldsymbol{w}_q^T$ is a row vector of weights applied to the bits before summation; these weights are also 0 or 1, the total number of nonzero weights in $\boldsymbol{w}_q$ being denoted $\#q$, the size of the subset of bits that this query interrogates. Clearly $A(q)$ is an integer (a *count*) in the range $\{0, \ldots \#q\}$. There are of course 2^n possible distinct queries.

A.1 Noiseless Reconstruction via Linear Algebra

Each noiseless query constitutes a linear constraint on the n bits, and distinct queries obviously constitute linearly independent constraints. Here "linear" and "independent" are used in the sense of linear algebra, which therefore guarantees that n independent queries are *sufficient* to reconstruct d. Since, however, each component of d (viewed as a vector in $\mathbb{R}^n$) is restricted to only two possible values, reconstruction may be possible with fewer than n queries.

In what follows, we will often speak of the "probability" of the value of a given bit or bits in the database. In the real world, the noiseless database is fixed, so its bits are not random variables. But in order to be able to apply information-theoretic arguments to the noiseless case, let's imagine that we are designing a reconstruction algorithm to be applied to the ensemble of *all possible* databases of n bits. In this ensemble, each bit takes on the values 0 or 1 with equal frequencies ($= 1/2$). To the extent that the actual database can be regarded as having been chosen "at random," the values of its bits can be regarded as independent random variables.

With this prolog, consider a reconstruction scheme in which we first query $n/2$ disjoint pairs of bits: e.g., the k^{th} query q_k interrogates bits $2k-1$ and $2k$, for $k \in \{1,\ldots,n/2\}$. In the average over all 2^n possible data bases, since each of the two bits interrogated is ± 1,

$$A(q_k) = \begin{cases} 0 & \text{with probability } 1/4, \\ 2 & \text{with probability } 1/4, \\ 1 & \text{with probability } 1/2 \end{cases}$$

When either of the first two possibilities is realized, both bits interrogated by q_k are determined. Thus we may expect to reconstruct $n/2$ of the bits with these $n/2$ queries—a plausible result! But, we now have partial information about the remaining $n/2$ bits that belong to "ambiguous" pairs where $A(q_k) = 1$: namely, the two bits of such a pair must be distinct. There will be approximately $n/4$ ambiguous pairs. Thus a further $\sim n/4$ queries that interrogate only the first member of each such pair will resolve the remaining ambiguities. By this argument, we may reconstruct the database with no more than $\sim 3n/4$ queries. This is fewer than would suffice by the linear-algebra argument, but not by much; which suggests that the linear-algebra argument, though not rigorous, may be useful. As we show in the following subsections, however, it may be possible to do still better—i.e. fewer queries needed for noiseless reconstruction—by a logarithmic factor.

A.2 Information: An Introductory Example

To further illustrate the point, take the simple case of a 3-bit database. Let (B_1, B_2, B_3) represent these bits, $B_i \in \{0,1\}$, each with probabilities $\Pr(B_i = 0) = \Pr(B_i = 1) = \frac{1}{2}$. Consider two queries, $Q_L = B_1 + B_2$ (which interrogates the two leftmost bits) and $Q_R = B_1 + B_2$. There are of course 8 possible databases, and three possible values for each query, as shown in Table A-4 below:

B_1	B_2	B_3	Q_L	Q_R
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	2
1	0	0	1	0
1	0	1	1	1
1	1	0	2	1
1	1	1	2	2

Table A-4: Two queries on a 3-bit database

All 8 rows are equally probable. The *entropy* of the joint distribution (probability mass function or PMF) of the three bits is therefore

$$H(B_1,B_2,B_3) = -\sum_{B_1,B_2,B_3} P(B_1,B_2,B_3)\log_2 P(B_1,B_2,B_3) = -8 \times \frac{1}{8}\log_2\frac{1}{8} = 3\,,$$

as one might expect. Notice that in 6 out of 8 cases, the values of the three bits are fully determined by the values of (Q_L, Q_R). The exceptions are those in which $Q_L = Q_R = 1$, there being two bit combinations 010 and 101 that give this result. So in $3/4$ of the cases, two queries suffice to determine the bits, while in the remaining $1/4$, a third query is needed. Thus the *average* number of queries needed to reconstruct the database is[4]

$$\frac{3}{4} \times 2 + \frac{1}{4} \times 3 = 2.25 \quad \text{queries on average}$$

[4]One might ask whether it's possible to do better with a different pair of initial queries. There are 28 possibile pairs [$2^3 \times (2^3 - 1)/2$], but none does better than this pair.

Another way to look at this is to say that in $3/4$ of the cases, the two queries yield 3 bits worth of information; while in the remaining $1/4$ of the cases, the queries leave one bit's worth of ambiguity (the choice between databases 010 and 101), so that then in effect they yield only 2 bits of information. Thus the average number of bits of information yielded by these two queries is

$$\frac{3}{4}\times 3+\frac{1}{4}\times 2 = 2.75 \quad \text{bits of information on average}$$

The joint PMF of (Q_L, Q_R), which follows from Table A-4, is

Q_L	Q_R	probability
0	0	1/8
0	1	1/8
1	0	1/8
1	1	2/8
0	2	0
2	0	0
1	2	1/8
2	1	1/8
2	2	1/8

Table A-5: Joint probability mass function of two queries.

The entropy of these two variables is therefore (combinations that have zero probability being omitted from the sum)

$$-\sum_{Q_L,Q_R} P(Q_L,Q_R)\log_2 P(Q_L,Q_R) = -6\times\frac{1}{8}\log_2\frac{1}{8} \; -\frac{1}{4}\log_2\frac{1}{4} = 2.75$$

Evidently, the entropy of the PMF of (Q_L, Q_R) coincides with the average number of bits of information gained from these two queries. This generalizes.

Looking ahead to Section A.4, the covariance of these two queries is

$$\boldsymbol{C} = \mathrm{cov}(Q_L, Q_R) = \frac{1}{4}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

and the Gaussian approximation described there predicts that

$$H(Q_L, Q_R) \approx \frac{1}{2}\log_2 \det(2\pi e \boldsymbol{C}) \approx 2.88667$$

This is an overestimate (2.88667 instead of 2.75), presumably because the Gaussian approximation is not accurate for queries involving small numbers of bits. Yet it is qualitatively correct: 2 well-chosen queries on 3 bits yield > 2 but < 3 bits of information on average.

A.3 Information Gained Per Query

In the examples above, why do we do better by querying two bits at a time, and how can this be generalized?

Querying a single bit—noiselessly—reaps exactly one bit of information, because there are two possible outcomes (0 or 1), and averaged over all possible databases, these outcomes have equal frequency.

Consider now a query q that sums $\#q = m \geq 1$ bits. There are now $m+1$ possible values for the answer $A(q) = a \in \{0, \ldots, m\}$. In the data-base ensemble, the probabilities or frequencies frequencies $\{f_a\}$ of these outcomes have the binomial distribution $B(m, 1/2)$, meaning that

$$f_a = 2^{-m}\binom{m}{a}, \qquad \Rightarrow \quad \sum_a f_a = 1. \tag{A-2}$$

The formal information gained from this query is then

$$I(A) = -\sum_a f_a \log_2 f_a \tag{A-3a}$$

$$\approx \tfrac{1}{2}\log_2 m + \underbrace{\tfrac{1}{2}\log_2(\pi e/2)}_{\approx 1.047096} \equiv I_G(A) \tag{A-3b}$$

The second line is obtained by approximating the binomial distribution as a Gaussian (with mean $E(A) = m/2$ and variance $m/4$). Table A-6 shows that the Gaussian approximation is quite good even for small m—but not for $m = 0$, a point that will be important in Section A.7.

m	I	I_G
0	0	$-\infty$
1	1	1.047096
2	3/2	1.547096
16	3.04655	3.047096
128	4.547088	4.547096

Table A-6: Average information gain, in bits, from a single noiseless query that sums m bits. Second column is exact; third column is the Gaussian approximation.

What we have called $I(m)$ is also the *entropy* $H(X)$ of a binomially distributed random variable $X \sim B(m, 1/2)$. We use the notation I rather than H in this instance because we think of it as measuring the average *knowledge gained* after a query, rather than the *uncertainty* in the outcome of the query. But regardless of the interpretation, the mathematical rules governing information/entropy are the same.

A.4 Information Gained from Multiple Noiseless Queries

The preceding discussion shows that the most informative *single* query is the sum of all n bits: the information gained is $I(n) \approx 0.5 \log_2(n)$ for $n \gg 1$. But of course this is not enough to reconstruct all $n \gg \log_2 n$ bits. Clearly reconstruction requires multiple queries; but what is the minimum number? One may speculate that since a single query q that sums $\#q \sim O(n)$ bits yields $O(\log n)$ bits of information, it should follow that the minimum number of such queries required is $O(n/\log n)$. But this is not obvious, because queries are not independent unless they interrogate disjoint subsets of the n bits. Therefore their information will not simply add. In the first two schemes above, the subsets *were* independent: those queries interrogated individual bits or disjoint pairs of bits. But such "small" queries [$\#q \sim O(1)$] yield less information (at least individually) than "large" queries [$\#q \gg 1$]. And for $n \gg 1$, since we will need *at least* $O(n/\log n)$ queries to reconstruct, they cannot be entirely disjoint if they are individually

large.

Consider now two queries q_1 and q_2, and let $q_1 \cap q_2$ be the subset of bits that they have in common. If these queries are large, i.e., $\min(\#q_1, \#q_2) \gg 1$, then by the Central Limit Theorem, they are well approximated as Gaussian random variables, with means $E(q_i) = \frac{1}{2}\#q_i$ for $i \in \{1,2\}$, and covariance matrix

$$\boldsymbol{C} = \frac{1}{4}\begin{pmatrix} \#q_1 & \#(q_1 \cap q_2) \\ \#(q_1 \cap q_2) & \#q_2 \end{pmatrix}$$

(The prefactor comes from the fact that the mean-subtracted bit values are $\pm\frac{1}{2}$, whence the variance of individual bits is $\frac{1}{4}$.) It is easily seen that if the "information" of a multivariate Gaussian density function

$$P(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \frac{1}{\sqrt{\det(2\pi\boldsymbol{C})}} \exp\left(-\frac{1}{2}\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{C}\boldsymbol{x}\right)\mathrm{d}\boldsymbol{x}$$

is defined by $-\int P(\boldsymbol{x}) \log_2 P(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$, then this information is

$$I(\boldsymbol{C}) = \log_2 \sqrt{\det(2\pi e\boldsymbol{C})}\,, \tag{A-4}$$

This reduces to the Gaussian approximation of Section A.3 for a single query, where $\boldsymbol{C} \to m/4$, a scalar. For multiple *disjoint* queries, so that $\boldsymbol{C}$ is diagonal, eq. (A-4) says that the total information is the sum of the informations gained from each query separately. If the queries are not disjoint, then at least some of the off-diagonal entries of $\boldsymbol{C}$ are positive, and none are negative, whence the determinant of $\boldsymbol{C}$ is less than the product of its diagonals: this means that the total information is less than the sum of the information obtained from the individual queries.

The goal now is to find the smallest rank r (i.e., the smallest number of queries) for which $I(\boldsymbol{C}) > n$, with the restriction that

$$\boldsymbol{C} = \frac{1}{4}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{W}\,, \tag{A-5}$$

for some $n \times r$ matrix $\boldsymbol{W}$ whose entries are 0 or 1: each column of $\boldsymbol{W}$ corresponds to a query vector $\boldsymbol{w}_q$. If the information $I(\boldsymbol{C}) > n$, we can expect to be able to reconstruct "most" n-bit databases with these r queries.

Suppose, to begin with, that the entries of the matrix $\boldsymbol{W}$ are chosen at random. In this case, approximately half of the elements in each column (i.e., in each query vector) would be 1, and the remainder 0; but the excess or deficit of 1s over 0s in each column would fluctuate by $O(\sqrt{n})$. Any two distinct columns of $\boldsymbol{W}$ would have approximately $n/4$ 1s in common, so that $\sum_k W_{ik}W_{kj} \approx (n/4)(1+\delta_{ij})$. The elements of the covariance matrix would then be

$$C_{ij} = \begin{cases} n/8 + O(\sqrt{n}) & \text{if } i = j \in \{1,\ldots,r\} \\ n/16 + O(\sqrt{n}) & \text{if } i \neq j \end{cases} \tag{A-6}$$

The $O(\sqrt{n})$ are random in sign and have mean 0, so that it might be hoped that in computing $\log_2 \det \boldsymbol{C}$ for sufficiently large n, we could neglect them compared to the $O(n)$ terms. The matrix with these terms neglected is

$$\bar{\boldsymbol{C}} = \frac{n}{16}\left(\boldsymbol{I}_r + \boldsymbol{J}_r\right), \tag{A-7}$$

in which $\boldsymbol{I}_r$ is the $r \times r$ identity matrix, and the matrix $\boldsymbol{J}_r$ is entirely filled with 1s (sometimes called the "unit" matrix, although this risks confusion with the identity). Since $\boldsymbol{I}_r$ commutes with $\boldsymbol{J}_r$, the two matrices can be simultaneously diagonalized, and their eigenvalues simply add.

It is not hard to see that the eigenvectors of $\boldsymbol{J}$ have the form

$$\boldsymbol{v}_\omega = (1, \omega, \omega^2, \ldots, \omega^{r-1})^T$$

with $\omega^r = 1$, i.e. ω is any of the r^{th} roots of unity. These eigenvectors are orthogonal ($\boldsymbol{v}_\omega^\dagger \boldsymbol{v}_{\omega'} = r\delta_{\omega,\omega'}$), as is familiar from the Discrete Fourier Transform. For the trivial root $\omega = 1$, the eigenvalue of $\boldsymbol{J}$ is r, while all of the $r-1$ other roots correspond to zero eigenvalues. Therefore the eigenvalues of $\boldsymbol{I} + \boldsymbol{J}$ are

$$\{\underbrace{1,\ldots,1}_{r-1 \text{ times}}, 1+r\},$$

and it follows that

$$\begin{aligned} I(\bar{\boldsymbol{C}}) &\equiv \tfrac{1}{2}\log_2 \det(2\pi e \bar{\boldsymbol{C}}) \\ &= \tfrac{1}{2} r \log_2\left(\frac{\pi e}{8} n\right) + \tfrac{1}{2}\log_2(1+r) \\ &\approx \tfrac{1}{2} r (\log_2 n + 0.094) \qquad \text{for } r, n \gg 1. \end{aligned} \tag{A-8}$$

A.5 m Sequences and Hadamard Matrices

The replacement

$$\boldsymbol{C} \to \bar{\boldsymbol{C}}$$

is an approximation. But we can obtain the determinant (A-7) exactly in the special cases that $n = 2^k - 1$ through a cunning *pseudo*random choice of the query vectors: namely, m-sequences, a.k.a. maximum-length Linear Feedback Shift Register (LFSR) sequences [11]. In the form we need them here, they are periodic sequences of bits $b_i \in \{0,1\}$ with period $n = 2^k - 1$ and autocorrelation function

$$A(j) \equiv \sum_{i=0}^{n-1} b_i b_{i+j} = \begin{cases} (n+1)/2 & \text{when } j \equiv 0 \mod n \\ (n+1)/4 & \text{otherwise} \end{cases} \tag{A-9}$$

If we populate the columns of $\boldsymbol{W}$ with distinct circular shifts of such a sequence, then $\boldsymbol{C}$ will have almost exactly the form (A-7), the only change being that $n \to n+1$ (an even number). Then the information gained from these r queries will be exactly as in the second line of (A-8), except for the same replacement.[5]

Hadamard matrices yield similarly good correlation properties [11]. By definition, a Hadamard matrix of order n is an $n \times n$ matrix $\boldsymbol{H}$ whose entries are ± 1 and whose rows are orthogonal, so that $\boldsymbol{H}\boldsymbol{H}^T = n\boldsymbol{I}$, where $\boldsymbol{I}$ is the $n \times n$ identity. The order n must be 1, 2, or a multiple of 4; it is conjectured but not proved that Hadamard matrices exist for every multiple of 4. There are explicit constructions for special cases, however, and in particular for $n = p + 1$ where p is a prime of the form $4k - 1$ (i.e. $n \in \{4, 8, 12, 20, 24, 32, 44, 48, 60, \ldots\}$). Importantly, the first row (and first column) of the latter sort[6] of Hadamard matrix is all 1s, so it follows from the definition that each of the remaining rows has an equal number of $+1$s and -1s. It is then not hard to see that if we replace the elements H_{ij} of such a matrix with

$$W_{ij} = \frac{1}{2}(H_{\sigma(j)i} + 1)\,,$$

[5] Exact, that is, within our Gaussian approximation for the binomial query outcomes.

[6] a "cyclic" Hadamard matrix [11]

so that the j^{th} column of $\boldsymbol{W}$ is the $\sigma(j)^{\text{th}}$ *row* of $\boldsymbol{H}$ with every -1 replaced by 0, then the elements of $\boldsymbol{W}^T\boldsymbol{W}$ are

$$\sum_{i=1}^{n} W_{ij}W_{ik} = \begin{cases} n & j=k \;\&\; \sigma(j)=1, \\ n/2 & j=k \;\&\; \sigma(j)\neq 1, \\ n/2 & j\neq k \;\&\; \min(\sigma(j),\sigma(k))=1, \\ n/4 & j\neq k \;\&\; \min(\sigma(j),\sigma(k))\neq 1. \end{cases} \tag{A-10}$$

Here $\sigma()$ is any permutation of $\{1,2,\ldots,n\}$ But we do not have to use the complete permutation: we can use a part of it that selects some subset of r rows from $\boldsymbol{H}$, in which case $\boldsymbol{W}$ becomes $n\times r$, while the covariance matrix $\boldsymbol{C}\equiv\frac{1}{4}\boldsymbol{W}^T\boldsymbol{W}$ becomes $r\times r$. If this subset does not include the first row of $\boldsymbol{H}$ (the row that is all 1s), then $\boldsymbol{C}$ has exactly the form (A-7), and hence the same eigenvalues and determinant. If the first row of $\boldsymbol{H}$ is included, then the eigenvalues and determinant can be found by Cholesky decomposition $\boldsymbol{C}=\boldsymbol{L}\boldsymbol{L}^T$, where $\boldsymbol{L}$ is lower triangular.

The diagonal entries of $\boldsymbol{L}$ are the square roots of the eigenvalues of $\boldsymbol{C}$. It turns out that when the first column of $\boldsymbol{W}$ is the first row of $\boldsymbol{H}$, the first diagonal of $\boldsymbol{L}$ is $\sqrt{n}/2$, all the rest are $\sqrt{n}/4$, and the rest of $\boldsymbol{L}$ vanishes except for the first column, in which all the elements after the first are also $\sqrt{n}/4$. In this case, all of the eigenvalues of $\boldsymbol{C}$ coincide with those of (A-7) (i.e., they are $n/16$) except for the first, which is $n/4$ in this case, but $n(r+1)/16$ in (A-7). So if $r<n$ (fewer queries than bits), it is slightly advantageous not to use the first row of $\boldsymbol{H}$, i.e. not to include the query that sums all of the bits.

A.6 The Minimal Number of Queries

We have seen that, within our Gaussian approximation at least, and neglecting $O(1)$ corrections, the information gained from $r\leq n$ noiseless queries on an n-bit database can be made as large as

$$\max(I_r)\approx\frac{r}{2}\left[\log_2 n+\log_2(\pi e/8)\right].$$

On the other hand, it follows from eq. (A-3a) that the maximum information obtained from a single query is $\max(I_1)\lesssim\log_2 n+\log_2(\pi e/2)$: we do best by sum-

ming all of the n bits. It would seem therefore that the redundancy among multiple queries can be made almost neglegible, i.e. $\max(I_r) \approx r\max(I_1)$: the information contributed by distinct queries is almost additive, apart from the different constants $\log_2(\pi e/2)$ vs. $\log_2(\pi e/8)$.

In the absence of prior constraints on the bits in the database, we must have $I_r \geq n$ in order to determine all of the bits. Thus

> *The minimum number of noiseless queries needed to reconstruct an n-bit database is at least $2n/\log_2 n$ for large n.*

We have tested this by numerical experiments with modest values of n and r, as shown in Table A-7. Using a modified hill-climbing technique, we have constructed a set of near-optimal (better than random) queries[7]. As shown in the fourth column, most of the 2^n possible databases answer our $\lceil 2n/\log_2 n \rceil$ queries uniquely, but not all. As we add queries, the number of ambiguous cases appears to drop exponentially. The third column shows the minimum number of queries needed to resolve all ambiguities. The evidence of this table suggests that the $r \sim 2n/\log_2 n$ criterion is relevant, but because exhaustion over all 2^n databases is impractical for much larger n, it is also consistent with the possibility that the minimum r/n needed to resolve all ambiguities asymptotes to a constant. This is what was found empirically in Section 4 but it's important to note that there is no guarantee that the least squares approach used there is optimal in the Shannon or information-theoretic sense.

A.7 Noisy Single Queries

Instead of the exact answer (A-1) to a query, we receive a noisy version $\hat{A}(q) = \boldsymbol{w}_q^T \boldsymbol{d} + N_q$, where N_q is a random variable independent of the database and query

[7] by attempting to maximize $\boldsymbol{W}^T\boldsymbol{W}$, with the restriction that $\boldsymbol{W}$ is $n \times r$ and its entries are all 0 or 1

n	$\lceil 2n/\log_2 n \rceil$	$r_{\min}$	uniques
8	6	6	98.4%
9	6	6	100%
10	7	7	100%
11	7	8	96.9%
12	7	9	88.7%
13	8	9	96.1%
14	8	9	94.6%
15	8	9	90.1%
16	8	10	83.5%
17	9	11	93.8%
18	9	13	88.0%
19	9	13	79.3%
20	10	14	95.8%
21	10	14	90.9%

Table A-7: Numerical experiments on noiseless queries of small databases. 2nd column is the smallest integer $\geq 2n/\log_2 n$. 3rd column is the minimum number of optimized queries needed to determine all 2^n databases uniquely. 4th is the fraction that are uniquely identified by $\lceil 2n/\log_2 n \rceil$ queries.

vectors. For convenience, the noise variables N_q and $N_{q'}$ belonging to distinct queries q and q' will be assumed independent and identically distributed.[8]

Presumably also there is a rule that a given query can be asked at most once—or if not, that the value taken by N_q is the same every time that query is asked: for if not, it would be possible to beat down the noise by asking the query repeatedly and averaging the answers.

The concept of *mutual information* $I(X,Y)$ is useful to express the knowledge that one has of a random variable X given an observation of a second variable Y, which for this application is a noisy version of X (Fig. A-1).

[8]This is not essential. In fact, the High Dimensional Matrix Method used by Census [19]) creates correlations among the N_q. As long as the noise remains independent of the database, the effect is to replace the noise covariance matrix $\sigma_N^2 \boldsymbol{I}$ in eq. (A-14) with some other (symmetric) matrix.

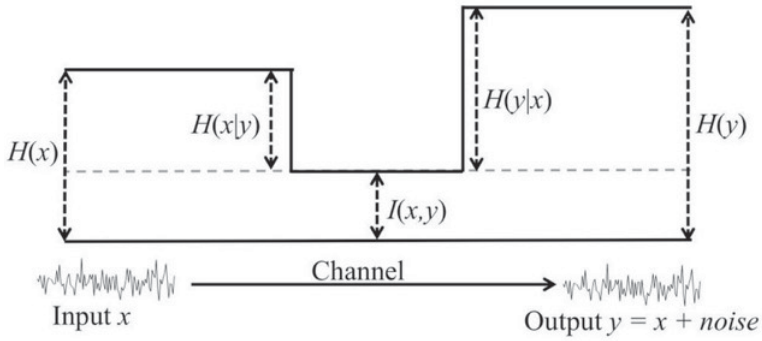


Figure A-1: Communication over a noisy channel. X ranges over transmitted signals, and Y over the noisy versions received. The entropy $H(X)$ is the minimum number of noiseless bits required to specify the value of X, and similarly for $H(Y)$. $H(X|Y)$ is the average uncertainty ($\sim$unknown bits) in X given a measurement of Y. The difference $I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$ is the mutual information.

The formal definition for discrete variables is

$$I(X;Y) = \sum_{X=x} \sum_{Y=y} p_{X,Y}(x,y) \log_2 \frac{p_{X,Y}(x,y)}{p_X(x) p_Y(y)} \,. \tag{A-11}$$

Here the sums are taken over all possible values x and y of X and Y respectively, while p_X, p_Y, and $p_{X,Y}$ are the probability mass functions (PMFs) for X alone, for Y alone, and for (X,Y) jointly. It can be shown that $I(X;Y) \geq 0$, with equality iff X and Y are independent.

A small example may increase confidence in this definition. Suppose X represents a single-bit message with equally frequent values $\{0,1\}$, and $Y = X + N$ with N a noise bit that is also equally likely to be 0 or 1. Therefore $Y \in \{0,1,2\}$. The PMFs are described by the following table:

x	y	$p_X(x)$	$p_Y(y)$	$p_{X,Y}(x,y)$
0	0	1/2	1/4	1/4
0	1	1/2	1/2	1/4
0	2	1/2	1/4	0
1	0	1/2	1/4	0
1	1	1/2	1/2	1/4
1	2	1/2	1/4	1/4

The third and fourth entries in the last column (for the joint PMF) vanish, because for example if $X = 0$ then $Y = 2$ is impossible, as the noise bit is at most 1. If $Y = 0$ or $Y = 2$, then X is determined (as 0 or 1, respectively). Taken together, these outcomes happen half the time: $p_{X,Y}(0,0) + p_{X,Y}(1,2) = 1/2$. In case $Y = 1$, however, X is equally likely to be 0 or 1. So observing Y yields perfect knowledge of X half the time, and the rest of the time no information at all. We may therefore say that observing Y is worth half a bit of knowledge about X on average. If one works through the definition (A-11) using the values in this table,[9] one finds indeed that $I(X;Y) = 1/2$.

A general theorem about mutual information is[22]

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)\,,$$

in which $H(X)$ and $H(Y)$ are the entropies[10] of X and Y separately, while $H(X|Y)$ is the residual entropy of X after Y is observed, and similarly for $H(Y|X)$. This is illustrated in Fig. A-1. It is easily seen that if X and N are independent, then $H(X+N|X) = H(N)$. Therefore,

$$I(X; X+N) = H(X+N) - H(N) \quad \text{when } X \text{ is independent of } N. \qquad \text{(A-12)}$$

Suppose for example that X and N are independent univariate Gaussian variables, so that $Y = X + N$ is also Gaussian, and $\mathrm{var}\,Y = \mathrm{var}\,X + \mathrm{var}\,N$. Since the

[9] It is understood that $0 \cdot \log_2 0 = 0$, i.e. cases for which $p_{X,Y}(x,y) = 0$ are excluded from the sum.

[10] See the discussion of entropy vs. information in Section A.3

entropy of a Gaussian is[11] $H(X) = \frac{1}{2}\log_2(2\pi e \operatorname{var} X)$, and similarly for $H(Y)$ and $H(N)$, it follows that

$$I(X;Y) = \tfrac{1}{2}\log_2\left(1 + \frac{\operatorname{var}(X)}{\operatorname{var}(N)}\right). \tag{A-13}$$

The logarithm here is strongly reminiscent of the factor $\log_2\left(1 + \frac{P_{\text{signal}}}{P_{\text{noise}}}\right)$ in Shannon's channel capacity theorem [31].

To relate this result to the previous discussion of noiseless queries, we need to understand what happens as the variance of the noise tends to zero. In this limit, the Gaussian approximation breaks down. The exact query results (X) are actually integers with a binomial distribution. If noise with $\operatorname{var}(N) \ll 1$ is added to such queries, the exact result (X) can be obtained from $X+N$ by rounding to the nearest integer with negligible probability of error. So we should expect $I(X, X+N)$ to reduce to $H(X)$, which is finite, as $\operatorname{var}(N) \to 0$. However, eq. (A-13) presumes that both X and N take real values, and it yields an infinite result as $\operatorname{var}(N) \to 0$ because arbitrarily close real numbers can always be distinguished.

Suppose instead that both X and N are discrete independent independent variables, for example with binomial distributions $B(m, 1/2)$ and $B(m', 1/2)$ respectively. Then $Y = X + N$ is distributed as $B(m+m', 1/2)$. Also[12] $\operatorname{var}(X) = m/4$, $\operatorname{var}(N) = m'/4$, and $\operatorname{var}(Y) = (m+m')/4$. If $m' \geq 1$, then the Gaussian approximations for $H(N)$ and $H(Y)$ are quite accurate, as shown by Table (A-6), so that eq. (A-13) is a good approximation to the mutual information. But in the noiseless case $m' = 0$, we have to use the exact definition in the first line of eq. (A-3a) for the entropy of a binomial; this yields $H(N) = 0$. Then it follows from eq. (A-12) that $I(X;Y) \to I(X;X) = H(X)$, as we expect, rather than $+\infty$ as the Gaussian approximation (A-13) would predict in the noiseless limit.

[11] For a multivariate Gaussian, this becomes $H(\boldsymbol{X}) = \frac{1}{2}\log_2 \det[2\pi e \operatorname{cov}(\boldsymbol{X})]$, where $\operatorname{cov}(\boldsymbol{X})$ is the covariance matrix of $\boldsymbol{X}$

[12] Recall that if $X \sim B(n,p)$, where p is the probability of "success" on a single trial and n is the number of trials, that $\operatorname{var}(X) = np(1-p)$.

A.8 Multiple Noisy Queries

This generalizes directly to multiple queries, represented by a vector $\boldsymbol{X}$ when exact, but corrupted by a noise vector $\boldsymbol{N}$ with diagonal covariance $\mathrm{cov}(\boldsymbol{N}) = \sigma_N^2 \boldsymbol{I}$. Provided $\sigma_N^2 \gtrsim 1/4$, we may use the Gaussian approximation, so that

$$I(\boldsymbol{X}, \boldsymbol{X}+\boldsymbol{N}) \approx \tfrac{1}{2} \log_2 \det[\sigma_N^{-2} \boldsymbol{C} + \boldsymbol{I}]. \tag{A-14}$$

in which $\boldsymbol{C} = \mathrm{cov}(\boldsymbol{X})$ is determined as before by the $n \times r$ query matrix $\boldsymbol{W}$ [eq. (A-5)], and $\boldsymbol{I}$ is the $r \times r$ identity.

The result (A-14) should be interpreted as the total information gathered by these queries in the presence of noise. As we've seen in Section A.4, for sensible (e.g. random) choices of the query matrix $\boldsymbol{W}$, all but one of the eigenvalues of $\boldsymbol{C}$ is approximately equal to $n/16$ if $n \geq r \gg 1$. It follows that the net information gathered on average is

$$I_{\mathrm{net}} \approx \frac{r-1}{2} \log_2 \left(1 + \frac{n}{16\sigma_N^2}\right) + \frac{1}{2} \log_2 \left(1 + \frac{n(r+1)}{16\sigma_N^2}\right). \tag{A-15}$$

(The second logarithm comes from the one nonzero eigenvalue of the matrix $\boldsymbol{J}$ discussed above.) If there is to be hope of reconstructing the database, the information I_{net} must be $\geq n$, the number of bits to be reconstructed. If the standard deviation of the noise $\sigma_N > \sqrt{n/48}$, however, then the logarithm < 2, in which case we will not have enough information even at $r = n$—i.e., even if we make as many queries as bits. This is reminiscent of DN's result to the effect that $O(\sqrt{n})$ noise is sufficient to prevent an "algebraically bounded" adversary from reconstructing the database.

But now suppose that we are allowed to make $r \gg n$ queries. This is most interesting in the large-noise limit, i.e. where σ_N^2 is large compared to all of the eigenvalues of $\boldsymbol{C}$. Note by the way that $\boldsymbol{C}$ becomes singular for $r > n$, because it is constructed from $\boldsymbol{W}$, which has rank $\min(r,n)$. However, the combination $\sigma_N^2 \boldsymbol{C} + \boldsymbol{I}$ is nonsingular, and for sufficiently large σ_N^2, the expansion

$$\log_e \det(\boldsymbol{I} + \varepsilon \boldsymbol{M}) \to \varepsilon \mathrm{Trace}(\boldsymbol{M}) + O(\varepsilon^2) \quad \text{as } \varepsilon \to 0 \text{ at fixed } \boldsymbol{M}$$

allows us to write

$$I_{\text{net}} \approx \frac{\log_2 e}{2\sigma_N^2}\text{Trace}(\boldsymbol{C}) \approx \frac{nr\log_2 e}{16\sigma_N^2} \quad (\sigma_N^2 \gg n/16) \tag{A-16}$$

Hence, even if the signal-to-noise ratio per query is very small, a sufficient number of queries—specifically, $r \gtrsim 16\sigma_N^2/\log_e 2$—should gather enough information to determine the database. We have not checked this prediction experimentally but we do confirm that it is possible to gather sufficient information to reconstruct the DN database provided we can issue enough queries. Note that this result indicates one will always recover the bits if the variance of the noise is held fixed as the queries are issued.

A.9 Reconstruction

So far we've talked about gathering enough information, through queries, to *determine* the bits in a database; but we haven't provided a method for actually estimating the bits from the query results. Methods based on bounded least squares optimization are discussed elsewhere in this report, and illustrated by numerical experiments. Here we provide an alternative approach, straightforwardly applying Bayesian inference to our Gaussian approximation. For simplicity, we discuss here only the noiseless case, but the method is easily generalized to include noise.

The general idea is this. We choose a full $n \times n$ matrix $\boldsymbol{W}$ of query weights, with $\det \boldsymbol{W}$ nonzero. We then ask, after the first $r < n$ of these queries (defined by the first r columns of $\boldsymbol{W}$) have been posed and answered, what is the posterior (conditional) probability distribution for the answers to the remaining $n-r$ queries that have not yet been made? If this posterior is narrow, the likely answers to the not-yet-asked queries can be predicted with probable errors less than unity (i.e., less than a bit). Then, from the results of only the first r queries, we may write down a shrewd estimate for the full $n \times n$ linear system discussed in Section A.1 and invert for the bits (rounding the real-valued answers to 0 or 1 as needed). If on the other hand the posterior is not narrow enough, we increase r (i.e., ask more

queries) until it is.

This procedure is in principle well-defined if the queries are treated exactly as discrete binomial variables. But unfortunately we do not know how to make the exact calculations except by brute force. So we resort to our Gaussian approximation. Let $\boldsymbol{X}_n$ be the full length-n vector of random variables for the outcomes of all n queries defined by some $n \times n$ weight matrix $\boldsymbol{W}_n$ with entries $\in \{0,1\}$ and $\det \boldsymbol{W}_n \neq 0$. In the Gaussian approximation, the joint distribution of $\boldsymbol{X}_n$ is determined by the means $\boldsymbol{\mu}_n = E(\boldsymbol{X}_n)$ and covariances $\boldsymbol{C}_n = E\left[\boldsymbol{X}_n - \boldsymbol{\mu}_n)(\boldsymbol{X}_n - \boldsymbol{\mu}_n)^T\right]$. As in Section A.4, since we assume uniform priors on all of the database bits (0 or 1 with equal probability), each component of $\boldsymbol{mu}_n$ equals one half the sum of the corresponding column of $\boldsymbol{W}_n$, while $\boldsymbol{C}_n = \frac{1}{4}\boldsymbol{W}_{\boldsymbol{n}}^T \boldsymbol{W}_{\boldsymbol{n}}$.

Now partition $\boldsymbol{X}_n$ into its first r components $\boldsymbol{X}_r$ and the remaining $n-r$ components $\boldsymbol{X}_{n-r}$, with corresponding partitions of the means and covariances:

$$\boldsymbol{\mu} = \left[\begin{array}{c} \boldsymbol{\mu}_r \\ \hline \boldsymbol{\mu}_{n-r} \end{array}\right], \qquad \boldsymbol{C}_n = \left[\begin{array}{c|c} \boldsymbol{C}_r & \boldsymbol{C}_{r,n-r} \\ \hline \boldsymbol{C}_{n-r,r} & \boldsymbol{C}_{n-r} \end{array}\right] \tag{A-17}$$

Here

$$\boldsymbol{C}_r = E(\boldsymbol{X}_r \boldsymbol{X}_r^T)$$

represents the $r \times r$ covariances of the components of $\boldsymbol{X}_r$ among themselves, and similarly for

$$\boldsymbol{C}_{n-r} = E(\boldsymbol{X}_{n-r} \boldsymbol{X}_{n-r}^T);$$

while

$$\boldsymbol{C}_{r,n-r} = E(\boldsymbol{X}_r \boldsymbol{X}_{n-r}^T)$$

and its transpose

$$\boldsymbol{C}_{n-r,r} = E(\boldsymbol{X}_{n-r} \boldsymbol{X}_r^T)$$

encode the $r \times (n-r)$ cross-correlations between the components of $\boldsymbol{X}_r$ and $\boldsymbol{X}_{n-r}$. As is well known,[13] the conditional probability $\Pr(\boldsymbol{X}_{n-r} | \boldsymbol{X}_r = \boldsymbol{x}_r)$ is itself

[13] see, e.g., the Wikipedia article "Multivariate normal distribution" and references therein

Gaussian, with means and covariances

$$\begin{aligned}\hat{\mu}_{n-r} &= \boldsymbol{\mu}_{n-r} + \boldsymbol{C}_{n-r,r}\boldsymbol{C}_r^{-1}(\boldsymbol{x}_r - \boldsymbol{\mu}_{n-r}) \\ \hat{\boldsymbol{C}}_{n-r} &= \boldsymbol{C}_{n-r} - \underbrace{\boldsymbol{C}_{n-r,r}\boldsymbol{C}_r^{-1}\boldsymbol{C}_{n-r,r}^T}_{\boldsymbol{Q}}\,.\end{aligned} \tag{A-18}$$

Since the matrix $\boldsymbol{Q}$ is positive semidefinite, it follows that $\det\hat{\boldsymbol{C}} \le \det\boldsymbol{C}_{n-r}$, with equality only if the cross correlations $\boldsymbol{C}_{n-r,r}$ vanish.

Importantly, the reduced covariance matrix $\hat{\boldsymbol{C}}$ for the unposed $n-r$ queries does not depend on the results ($\boldsymbol{X}_r = \boldsymbol{x}_r$) of the first r queries, so we can work it out in advance in terms of the query weights $\boldsymbol{W}_n$. This can be done explicitly when $\boldsymbol{C}_n$ has the simple form (A-7), which we can obtain by choosing the columns of $\boldsymbol{W}$ to be m sequences, or by choosing them at random and neglecting the resulting $O(\sqrt{n})$ "fluctuations" in the resulting components of $\boldsymbol{C}$ [eq. (A-8)]. In this case, $\boldsymbol{C}_r$ and $\boldsymbol{C}_{n-r}$ have similar forms, except that in each case, $\boldsymbol{I}$ and $\boldsymbol{J}$ are matrices of the appropriate order.[14] It's clear that $\boldsymbol{J}_k^2 = k\boldsymbol{J}_k$ for every k, and therefore

$$(\boldsymbol{I}_k + \boldsymbol{J}_k)^{-1} = \boldsymbol{I}_k - \frac{1}{k+1}\boldsymbol{J}_k$$

The off-diagonal matrix $\boldsymbol{C}_{r,n-r} = \frac{n}{16}\boldsymbol{J}_{r,n-r}$, $\boldsymbol{J}_{k,m}$ being the $k \times m$ matrix with all entries equal to 1 (so that $\boldsymbol{J}_{k,k} = \boldsymbol{J}_k$). By means of the rules

$$\boldsymbol{J}_{j,k}\boldsymbol{I}_k = \boldsymbol{J}_{j,k} \quad \text{and} \quad \boldsymbol{J}_{i,k}\boldsymbol{J}_{k,j} = k\boldsymbol{J}_{i,j}$$

we can now evaluate the reduced covariance (A-18) for this choice of queries:

$$\hat{\boldsymbol{C}}_{n-r} = \frac{n}{16}\left(\boldsymbol{I}_{n-r} + \frac{1}{r+1}\boldsymbol{J}_{n-r}\right). \tag{A-19}$$

The determinant of $\hat{\boldsymbol{C}}_{n-r}$ is smaller than that of $\boldsymbol{C}_{n-r} = \frac{n}{16}(\boldsymbol{I}_{n-r} + \boldsymbol{J}_{n-r})$ by a factor $(2r+1)/(r+1)^2 \approx 2r^{-1}$ for $r \gg 1$. In logarithmic terms, this is a disappointingly slight reduction in uncertainty.

[14] I.e., $\boldsymbol{C}_k = \frac{n}{16}(\boldsymbol{I}_k + \boldsymbol{J}_k)$, with $\boldsymbol{I}_k$ being the $k \times k$ identity, and $\boldsymbol{J}_k$ being the $k \times k$ matrix with all elements equal to 1. The prefactor $\frac{n}{16}$ in $\boldsymbol{C}_k$, however, is invariant.

B MATLAB CODE FOR DN DATABASE RECONSTRUCTION

The MATLAB codes in this appendix can be used to generate the various figures in the report associated with the calculations on the Dinur-Nissim database.

Listing 1: Matlab script for Figure 5-1

```
% script to recover the bits in a Dinur–Nissim database without noise addition

max_n_data = 1000;
min_n_data = 1000;
step_n_data = 10;

% number of random trials

n_trials = 100;

n_entry = floor((max_n_data–min_n_data)/step_n_data)+1;

n_q_recovery = zeros(1,n_entry);
n_d = zeros(1,n_entry);
n_q_norm = zeros(1,n_entry);

completion_counter_max = 10;% the consecutive number of times the min fraction correct is 1 before terminating the queryloop

i_noise  = false; % set to false for no noise addition

i_entry = 0;

i_fig = 0;


for n_data = min_n_data:step_n_data:max_n_data

    % noise level – we add gaussian noise with mean 0 and variance eta

    sigma = sqrt(n_data)/2.0; % sigma for binomial distribution

    eta = sigma*log(n_data); % ensuring the noise is just above the sqrt(n) growth
```

```
33
34
35      % query_fraction = linspace(1/n_data,1.0,query_max);
36
37      % generate random data set
38
39      d = randi([0,1],n_data,1);
40
41      options = optimset('display','off'); % turn off the display
42
43      % set the lower and upper bounds on the solution
44
45      lb = zeros(n_data,1);
46      ub = ones(n_data,1);
47
48      fraction_correct = zeros(n_trials,10000);
49
50      i_query = 0;
51
52      completion_counter = 0;
53
54      while (completion_counter < completion_counter_max)
55
56          i_query = i_query + 1;
57
58          max_fraction_corrrect = 0.0;
59          max_residual_norm = 0.0;
60
61          for i_trial = 1:n_trials
62
63              % generate the random query matrix
64
65              Q = randi([0,1], i_query, n_data);
66
67              % generate the query answers
68
69              ans_q = Q*d;
70
71              % add noise to the answers
72
73              rand_vec = normrnd(0,eta, [i_query, 1]);
74
75              if (i_noise)
76                  ans_q = ans_q + rand_vec;
77              end
```

```
78
79              % now use constrained least squares to generate solution
80
81              [x_sol,res_norm,residual,exitflag,output] = lsqlin(Q,
                    ans_q,[],[],[],[],lb,ub, [], options);
82
83              max_residual_norm = max(max_residual_norm, res_norm);
84
85              % now round to 0 or 1
86
87              x_sol = round(x_sol);
88
89              % compute the percentage of bits returned correctly
90
91              n_correct = 0;
92
93              for i_bit = 1:n_data
94                  if (abs(x_sol(i_bit) - d(i_bit)) <= 1.0e-3)
95                      n_correct = n_correct +1;
96                  end
97              end
98
99              fraction_correct(i_trial, i_query)  = n_correct/n_data;
100
101         end
102
103         max_fraction_correct = max(fraction_correct(:,i_query));
104         min_fraction_correct = min(fraction_correct(:,i_query));
105
106         if ((min_fraction_correct - 0.9) >= 0)
107             completion_counter = completion_counter + 1;
108         else
109             completion_counter = 0;
110         end
111
112         fprintf (' %5i  trials  n_data: %5i query: %5i comp_counter:
                %5i  min_fraction_correct %8.4e max_frac_correct %8.4e
               max_residual: %8.4e \n', ...
113             n_trials, n_data, i_query, completion_counter,
                    min_fraction_correct, max_fraction_correct,
                    max_residual_norm)
114
115     end
116
117     n_query = i_query;
```

```
118
119     % now compute the mean percent correct and its variance
120
121     mean_fraction_correct = mean(fraction_correct);
122     var_fraction_correct = var(fraction_correct);
123
124     % now find the least value of query number that provides 100
            percent recovery
125
126     i_entry = i_entry+1;
127
128     n_d(i_entry) = n_data;
129
130     n_q_recovery(i_entry) = n_query;
131
132     for i = n_query:-1:1
133         if (abs(mean_fraction_correct(i) - 1) >= 1.0e-3)
134             n_q_recovery(i_entry) = i;
135             break;
136         end
137     end
138
139     % now produce a shaded distribution plot
140
141     x = 1:i_query;
142     y_mean = mean_fraction_correct(1:n_query);
143     y_10 = quantile(fraction_correct,0.10);
144     y_50 = quantile(fraction_correct,0.50);
145     y_90 = quantile(fraction_correct,0.90);
146
147     y_10 = y_10(1:n_query);
148     y_50 = y_50(1:n_query);
149     y_90 = y_90(1:n_query);
150
151
152     i_fig = i_fig+1;
153     figure(i_fig);
154     clf;
155
156     fprintf(' plotting figure %d...', i_fig);
157     hold on
158     plot(x,y_mean,'LineWidth',1.5);
159     plot(x,y_10);
160     plot(x,y_50);
161     plot(x,y_90);
```

```
    hold off
    title(['fraction correct vs. query for ', num2str(n_data),' bits
        with ',num2str(n_trials),' trials']);
    drawnow;
    fprintf (' plot complete\n')




end


% plot the min number of queries vs number of bits

i_fig = i_fig+1;

figure(i_fig);
clf;

plot (n_d(1:i_entry), n_q_recovery(1:i_entry));

drawnow;

% play with some possible normalizations of the min number of queries

for i_e = 1:i_entry
    n_q_norm(i_e) = n_q_recovery(i_e)/n_d(i_e);
    %    n_q_norm(i_e) = n_q_recovery(i_e)/n_d(i_e);
end

i_fig = i_fig+1;
figure(i_fig);
clf;

plot(n_d(1:i_entry), n_q_norm(1:i_entry));
```

Listing 2: Matlab script for Figures 5-2 and 5-3

```
% script to try to recover binary data set

max_n_data = 1000;
n_q_recovery = zeros(1,max_n_data);
n_d = zeros(1,max_n_data);
n_q_norm = zeros(1,max_n_data);

i_entry = 0;

for n_data = 100:100:max_n_data


    max_query = n_data;
    n_trials = 100;
    query_percent = linspace(1/n_data,1.0,max_query);

    % generate random data set

    d = randi([0,1],n_data,1);

    options = optimset('display','off'); % turn off the display

    % set the lower and upper bounds on the solution

    lb = zeros(n_data,1);
    ub = ones(n_data,1);

    percent_correct = zeros(n_trials,max_query);


    for i_query = 1:1:max_query

       fprintf (' n_data = %d   Performing query %d  with %d trials \
          n', n_data, i_query, n_trials)


        for i_trial = 1:n_trials

            % generate the random query matrix

            Q = randi([0,1], i_query, n_data);

            % generate the query answers

```

```
44            ans_q = Q*d;
45
46            % now use constrained least squares to generate solution
47
48            [x_sol,res_norm,residual,exitflag,output] = lsqlin(Q,
                  ans_q,[],[],[],[],lb,ub, [], options);
49
50
51            % now round to 0 or 1
52
53            x_sol = round(x_sol);
54
55            % compute the percentage of bits returned correctly
56
57            n_correct = 0;
58
59            for i_bit = 1:n_data
60                if (abs(x_sol(i_bit) - d(i_bit)) <= 1.0e-3)
61                    n_correct = n_correct +1;
62                end
63            end
64
65            percent_correct(i_trial, i_query)  = n_correct/n_data;
66
67        end
68
69    end
70
71    % now compute the mean percent correct
72
73    min_percent_correct = min(percent_correct);
74    mean_percent_correct = mean(percent_correct);
75    var_percent_correct = 2.0*var(percent_correct); % note I'm taking
          2 std devs
76    max_percent_correct = max(percent_correct);
77
78    % now find the lowest value of the number of queries that
        provides 100 percent recovery
79
80    i_entry = i_entry+1;
81
82    n_d(i_entry) = n_data;
83
84    n_q_recovery(i_entry) = max_query;
85
```

```
    for i = max_query:-1:1
        if (abs(mean_percent_correct(i) - 1) >= 1.0e-3)
            break;
        else
            n_q_recovery(i_entry) = n_q_recovery(i_entry) - 1;
        end
    end

    % plot error bar plot

    figure;

    errorbar (mean_percent_correct, var_percent_correct)



end

% plot the min number of queries vs number of bits

figure;

plot (n_d(1:i_entry), n_q_recovery(1:i_entry));

% play with some possible normalizations of the min number of queries -
% here we try direct proportionality to number of bits

for i_e = 1:i_entry
    n_q_norm(i_e) = n_q_recovery(i_e)/n_d(i_e);
end

figure;

plot(n_d(1:i_entry), n_q_norm(1:i_entry));
```

Listing 3: Matlab script for Figure 5-4

```
% script to examine the distribution of number of bits recovered for a
% fixed number of random bits in a database

max_n_data = 10;
min_n_data = 100;
step_n_data = 10;

% number of random trials

n_trials = 100;

n_entry = floor((max_n_data-min_n_data)/step_n_data)+1;

n_q_recovery = zeros(1,n_entry);
n_d = zeros(1,n_entry);
n_q_norm = zeros(1,n_entry);

completion_counter_max = 10;% the consecutive number of times the min fraction correct is 1 before terminating the queryloop

i_noise  = true; % set to false for no noise addition

i_entry = 0;

i_fig = 0;


for n_data = min_n_data:step_n_data:max_n_data

    % noise level - we add gaussian noise with mean 0 and variance eta

    sigma = sqrt(n_data)/2.0; % sigma for binomial distribution

    eta = sigma*log(n_data); % ensuring the noise is just above the sqrt(n) growth


    % generate random data set

    d = randi([0,1],n_data,1);

    options = optimset('display','off'); % turn off the display
```

```
41
42      % set the lower and upper bounds on the solution
43
44      lb = zeros(n_data,1);
45      ub = ones(n_data,1);
46
47      fraction_correct = zeros(n_trials,10000);
48
49      i_query = 0;
50
51      completion_counter = 0;
52
53      while (completion_counter < completion_counter_max)
54
55          i_query = i_query + 1;
56
57          max_fraction_corrrect = 0.0;
58          max_residual_norm = 0.0;
59
60          for i_trial = 1:n_trials
61
62              % generate the random query matrix
63
64              Q = randi([0,1], i_query, n_data);
65
66              % generate the query answers
67
68              ans_q = Q*d;
69
70              % add noise to the answers
71
72              rand_vec = normrnd(0,eta, [i_query, 1]);
73
74              if (i_noise)
75                  ans_q = ans_q + rand_vec;
76              end
77
78              % now use constrained least squares to generate solution
79
80              [x_sol,res_norm,residual,exitflag,output] = lsqlin(Q,
                    ans_q,[],[],[],[],lb,ub, [], options);
81
82              max_residual_norm = max(max_residual_norm, res_norm);
83
84              % now round to 0 or 1
```

```
85
86          x_sol = round(x_sol);
87
88          % compute the percentage of bits returned correctly
89
90          n_correct = 0;
91
92          for i_bit = 1:n_data
93              if (abs(x_sol(i_bit) - d(i_bit)) <= 1.0e-3)
94                  n_correct = n_correct +1;
95              end
96          end
97
98          fraction_correct(i_trial, i_query)  = n_correct/n_data;
99
100     end
101
102     max_fraction_correct = max(fraction_correct(:,i_query));
103     min_fraction_correct = min(fraction_correct(:,i_query));
104
105     if ((min_fraction_correct - 0.9) >= 0)
106         completion_counter = completion_counter + 1;
107     else
108         completion_counter = 0;
109     end
110
111     fprintf (' %5i  trials  n_data: %5i query: %5i comp_counter:
            %5i  min_fraction_correct %8.4e max_frac_correct %8.4e
           max_residual: %8.4e \n', ...
112         n_trials, n_data, i_query, completion_counter,
              min_fraction_correct, max_fraction_correct,
              max_residual_norm)
113
114 end
115
116 n_query = i_query;
117
118 % now compute the mean percent correct and its variance
119
120 mean_fraction_correct = mean(fraction_correct);
121 var_fraction_correct = var(fraction_correct);
122
123 % now find the least value of query number that provides 100
      percent recovery
124
```

```
    i_entry = i_entry+1;

    n_d(i_entry) = n_data;

    n_q_recovery(i_entry) = n_query;

    for i = n_query:-1:1
        if (abs(mean_fraction_correct(i) - 1) >= 1.0e-3)
            n_q_recovery(i_entry) = i;
            break;
        end
    end

    % now produce a shaded distribution plot

    x = 1:i_query;
    y_mean = mean_fraction_correct(1:n_query);
    y_10 = quantile(fraction_correct,0.10);
    y_50 = quantile(fraction_correct,0.50);
    y_90 = quantile(fraction_correct,0.90);

    y_10 = y_10(1:n_query);
    y_50 = y_50(1:n_query);
    y_90 = y_90(1:n_query);


    i_fig = i_fig+1;
    figure(i_fig);
    clf;

    fprintf(' plotting figure %d...', i_fig);
    hold on
    plot(x,y_mean,'LineWidth',1.5);
    plot(x,y_10);
    plot(x,y_50);
    plot(x,y_90);
    hold off
    title(['fraction correct vs. query for ', num2str(n_data),' bits
        with ',num2str(n_trials),' trials']);
    drawnow;
    fprintf (' plot complete\n')
```

```
end


% plot the min number of queries vs number of bits

i_fig = i_fig+1;

figure(i_fig);
clf;

plot (n_d(1:i_entry), n_q_recovery(1:i_entry));

drawnow;

% play with some possible normalizations of the min number of queries

for i_e = 1:i_entry
    n_q_norm(i_e) = n_q_recovery(i_e)/n_d(i_e);
    %    n_q_norm(i_e) = n_q_recovery(i_e)/n_d(i_e);
end

i_fig = i_fig+1;
figure(i_fig);
clf;

plot(n_d(1:i_entry), n_q_norm(1:i_entry));
```

Listing 4: Matlab script for Figure 6-6

```
% script to examine the accuracy of a sum query as a function of the
    value
% of epsilon

n_data_row = [100 200 500 1000 2000 5000];

% number of random trials

n_trials = 1000;

trial_result = zeros(n_trials,1);

% the set of privacy loss parameters we wish to examine

eps_row = [0.001 0.005 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09
    0.1 0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18 0.19  0.2 0.3 0.4 0.5
    0.6 0.7 0.8 0.9 1.0 ];


n_d_entry = length(n_data_row);
n_e_entry = length(eps_row);


query_accuracy = zeros(n_d_entry, n_e_entry); %


for i_d_entry = 1:n_d_entry  % loop over the values of the number of
    bits

    n_data = n_data_row(i_d_entry);

    fprintf (' number of data bits: %d \n ', n_data);

    for i_e_entry = 1:n_e_entry % loop over the values of epsilon

        epsilon = eps_row(i_e_entry);

        % noise level - we add gaussian noise with mean 0 and
            variance eta

        eta =  2/epsilon^2; % this sets the variance to the
            equivalent of the two sided exponential

        for i_trial = 1:n_trials % do a number of trials to get
```

```
            reasonablre statistics
39
40          % generate random data set
41
42          d = randi([0,1],n_data,1);
43
44          % compute the correct sum
45
46          sum_query = sum(d);
47
48          % add noise to the sum of the data set - here we add a
                Laplace
49          % distribution with parameter epsilon
50
51          unif = rand() - 0.5;
52          laplace_rand_var = -1./epsilon*sign(unif)*log(1-2*abs(
                unif));
53
54  %           rand_num = normrnd(0,sqrt(eta), [1, 1]);Q_n
55
56          rand_num  =  laplace_rand_var;
57          noised_sum = round(sum_query + rand_num);
58
59          trial_result(i_trial)  = 1.0 - abs((noised_sum-sum_query)
                /sum_query); % accuray - 1 is perfect and then it
                decreases as error decreases
60
61      end
62      mean_error = mean(trial_result);
63
64      fprintf ('      epsilon = %d variance = %d mean_error=%d\n',
            epsilon, eta, mean_error);
65
66      query_accuracy(i_d_entry,i_e_entry)  = mean_error;
67
68  end
69
70  end
71
72
73
74  % now plot the results
75
76  figure;
77
```

```
hold on

for i_curve = 1:n_d_entry

    x = eps_row;

    y = query_accuracy(i_curve, 1:n_e_entry);

    plot (x,y);

end

% set the axes — anything below a query accuracy of 0.0 is pretty
    useless
axis([0 1.0 0 1.01]);

% form the legend

for i_curve = 1:n_d_entry
    legendCell{i_curve} = num2str(n_data_row(i_curve), 'N =%-d');
end

legend(legendCell);

% label the axes

xlabel('Privacy loss parameter  — \epsilon');
ylabel('Query accuracy');

% title the plot

title(' Dinur–Nissim query accuracy vs privacy loss parameter \
    epsilon');
```

Listing 5: Matlab script for Figure 6-7

```
1
2  % Matlab script to examine what percentage of bits are recovered for
       a given
3  % privacy loss parameter and a given number of queries in the
       presence of
4  % noise. We use a two-sided Laplace distribution to sample the noise.
5
6  % the set of database size we wish to examine
7
8  n_data_row = [4000];
9
10 % number of random trials
11
12 n_trials = 10;
13
14 trial_fraction_correct = zeros(n_trials,1);
15
16 % the set of privacy loss parameters we wish to examine
17
18 eps_row = [ 0.01 0.02 0.03 0.04 0.05 0.1 0.2 0.25 0.3 0.4 0.5 1.0  ];
19
20 % the set of multiples of the number of data points we have that we
       wish to examine
21
22 n_mult_row = [1 5 10 20];
23
24 n_d_entry = length(n_data_row);
25 n_e_entry = length(eps_row);
26 n_m_entry = length(n_mult_row);
27
28 options = optimset('display','off'); % turn off the display for the
       optimizer
29
30 % array of fraction of number of bits correct as a function of number
        of bits, number of queries, and epsilon
31 fraction_correct = zeros(n_d_entry, n_m_entry,  n_e_entry);
32
33 % loop over the values of the number of bits
34 for i_d_entry = 1:n_d_entry
35
36     n_data = n_data_row(i_d_entry);
37
38     fprintf (' number of data bits: %d \n ', n_data);
39
```

```
40      % set the lower and upper bounds on the solution
41
42      lb = zeros(n_data,1);
43      ub = ones(n_data,1);
44
45      % generate random data set
46
47      d = randi([0,1],n_data,1);
48
49      % loop over the values of epsilon
50      for i_e_entry = 1:n_e_entry
51
52          epsilon = eps_row(i_e_entry);
53
54          % noise level - we add Laplce noise with mean 0 and variance
                eta
55          % this sets the variance to the equivalent of the two sided
                exponential
56          eta =  2/epsilon^2;
57
58          fprintf ('      epsilon = %d variance = %d \n', epsilon, eta)
                ;
59
60          % loop over the queries - we do various multiples of the
                number of
61          % data points
62
63          for i_m_entry = 1:n_m_entry
64
65              i_query = n_data*n_mult_row(i_m_entry);
66
67              % we do n_trials trials and average the results
68
69              max_residual_norm = 0;
70
71              for i_trial = 1:n_trials
72
73                  % generate the random query matrix
74
75                  Q = randi([0,1], i_query, n_data);
76
77                  % generate the query answers
78
79                  ans_q = Q*d;
80
```

```
81              % add noise to the answers
82
83              % add noise to the sum of the data set - here we add
                    a Laplace
84              % distribution with parameter epsilon
85
86              unif = rand(i_query,1) - 0.5;
87              laplace_rand_var = -1./epsilon.*sign(unif).*log(1-2*
                    abs(unif));
88
89              ans_q = ans_q + laplace_rand_var;
90
91              % now use constrained least squares to generate
                    solution
92
93              [x_sol,res_norm,residual,exitflag,output] = ...
94                  lsqlin(Q,ans_q,[],[],[],[],lb,ub, [], options);
95
96              max_residual_norm = max(max_residual_norm, res_norm);
97
98              % now round to 0 or 1
99
100             x_sol = round(x_sol);
101
102             % compute the percentage of bits returned correctly
103
104             n_correct = 0;
105
106             for i_bit = 1:n_data
107                 if (abs(x_sol(i_bit) - d(i_bit)) <= 1.0e-3)
108                     n_correct = n_correct +1;
109                 end
110             end
111
112             trial_fraction_correct(i_trial)  = n_correct/n_data;
113
114         end
115
116         max_fraction_correct = max(trial_fraction_correct);
117         min_fraction_correct = min(trial_fraction_correct);
118         mean_fraction_correct = mean(trial_fraction_correct);
119         var_fraction_correct = var(trial_fraction_correct);
120
121         fprintf ('              n_data: %5i query: %5i
                mean_fraction_correct %8.4e   max_residual: %8.4e \n',
```

```
                ...
122             n_data, i_query, mean_fraction_correct,
                    max_residual_norm)
123
124         fraction_correct(i_d_entry,i_m_entry,i_e_entry)  =
                mean_fraction_correct;
125
126     end
127 end
128
129 % now plot the results
130
131 [X, Y] = meshgrid(n_mult_row, eps_row);
132
133 % loop over the size of the data vector
134
135 Z = zeros(n_e_entry, n_m_entry);
136
137 for i_d_entry = 1:n_d_entry
138
139     for i_e_entry = 1:n_e_entry
140
141         for i_m_entry = 1:n_m_entry
142
143             Z(i_e_entry, i_m_entry) = fraction_correct(i_d_entry,
144                 i_m_entry, i_e_entry); % load the array of results for
                     each data set size
145         end
146
147     end
148
149     figure;
150     surf(X,Y,Z);
151     set(gca,'XScale','linear')
152     set(gca,'YScale','linear')
153
154 end
```

References

[1] John M Abowd. Staring Down the Database Reconstruction Theorem. Presentation to AAAS Annual Meeting Feb 16, 2019, 2019.

[2] Robert Ashmead. Estimating the Variance of Complex Differentially Private Algorithms. Presentation to Joint Statistical Meetings, American Statisticl Association, July 27, 2019, 2019.

[3] Raj Chetty and John Friedman. A practical method to reduce privacy loss when disclosing statistics based on small sample. *American Economic Review Papers and Proceedings*, 109:414–420.

[4] Graham Cormode, Tejas Kulkarni, and Divesh Srivastava. Constrained Differential Privacy for Count Data. *arXiv e-prints*, page 1710.00608, Oct 2017.

[5] Irit Dinur and Kobbi Nissim. Revealing Information while Preserving Privacy. In *PODS*, pages 202–210. ACM, 2003.

[6] Cynthia Dwork and Jing Lei. Differential Privacy and Robust Statistics. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pages 371–380. Association for Computing Machinery, 2009.

[7] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag.

[8] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Found. Trends. Theor. Comput. Sci.*, 9(3-4):211–407, 2013.

[9] Ivan P. Fellegi and Alan B. Sunter. A Theory for Record Linkage. *J. Am. Stat. Assoc.*, 64(328):1183–1210, 1969.

[10] Simson Garfinkel, John M. Abowd, and Christian Martindale. Understanding database reconstruction attacks on public data. *Commun. ACM*, 62(3):46–53, 2019.

[11] Solomon W. Golomb and Guang Gong. *Signal design for good correlation*. Cambridge, 2005.

[12] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2019.

[13] Mark Hansen. To Reduce Privacy Risk, Census Plans to Report Less Accurate Data. *New York Times*, Dec. 6, 2018.

[14] D. Kifer. Design Principles of the TopDown Algorithm. Presentation to JASON.

[15] Ios Kotsogiannis, Yuchao Tao, Ashwin Machanavajjhala, Gerome Miklau Umass, and Amherst Michael Hay. Architecting a Differentially Private SQL Engine. `http://cidrdb.org/cidr2019/papers/p125-kotsogiannis-cidr19.pdf`.

[16] Philip Leclerc. Generating Microdata with Complex Invariants under Differential Privacy. Presentation to Joint Statistical Meeting, American Statistical Association, 2019.

[17] Philip Leclerc. Results from a Consolidated Database Reconstruction and Intruder Re-identification Attack on the 2010 Decennial Census. Presentation at Workshop "Challenges and New Approaches for Protecting Privacy in Federal Statistical Programs", 2019.

[18] Justin Levitt. Uses of 2020 Redistricting Data. Presentation to JASON.

[19] Chao Li, Michael Hay, Gerome Miklau, and Yue Wang. A Data- and Workload-Aware Algorithm for Range Queries Under Differential Privacy. `http://arxiv.org/abs/1410.0265`, 2014.

[20] Chao Li, Gerome Miklau, Michael Hay, Andrew McGregor, and Vibhor Rastogi. The matrix mechanism: optimizing linear counting queries under differential privacy. *VLDB J.*, 24(6):757–781, 2015.

[21] Ashwin Machanavajjhala. Interpreting Differential Privacy. Presentation to JASON.

[22] David J. C. MacKay. *Information Theory, Inference, & Learning*. Cambridge University Press, 2003.

[23] Rachel Marks. How the 2020 Census Products Reflect Data user Feedback. Presentation to JASON.

[24] Laura Mckenna. Disclosure Avoidance for the 1970-2010 Censuses, 2018. https://www2.census.gov/ces/wp/2018/CES-WP-18-47.pdf.

[25] Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. Optimizing Error of High-dimensional Statistical Queries Under Differential Privacy. *Proc. VLDB Endow.*, 11(10):1206–1219, June 2018.

[26] Kobbi Nissim, Thomas Steinke, Alexandra Wood, Micah Altman, Aaron Bembenek, Mark Bun, Marco Gaboardi, David R O'brien, and Salil Vadhan. Differential privacy: a primer for a non-technical audience. *Vanderbilt J. Entertain. Technol. Law*, page 1021596, 2018.

[27] A. Ramachandran, L. Singh, E. Porter, and F. Nagle. Exploring Re-Identification Risks in Public Domains. In *2012 Tenth Annual International Conference on Privacy, Security and Trust*, 2012.

[28] Jerome P. Reiter. Differential Privacy and Federal Data Releases. *Annu. Rev. Stat. Its Appl.*, 6(1):85–101, 2018.

[29] Steven Ruggles, Catherine Fitch, Diana Magnuson, and Jonathan Schroeder. Differential Privacy and Census Data: Implications for Social and Economic Research. *AEA Pap. Proc.*, 109:403–408, 2019.

[30] William Sexton. Disclosure Avoidance At-Scale. Presentation to JASON.

[31] C. E. Shannon. Communication in the presence of noise. *Proc. Inst. Radio Engineers*, 37(1):10–21, 1949.

[32] Latanya Sweeney, Merce Crosas, and Michael Bar-Sinai. Sharing Sensitive Data with Confidence: the Datatags System. *Technol. Sci.*, pages 1–34, 2015.

[33] US Census Bureau. Census Bureau Continues to Boost Data Safeguards. https://www.census.gov/newsroom/blogs/random-samplings/2019/07/boost-safeguards.html.

[34] US Census Bureau. Census End to End Disclosure Avoidance System. https://github.com/uscensusbureau/census2020-das-e2e, 2019.

[35] US Census Bureau. Census Population Density by County. https://www.census.gov/library/visualizations/2010/geo/population-density-county-2010.html, 2019.

[36] D. van Riper. Differential Privacy and the Decennial Census. Presentation to JASON.

[37] David van Riper. Differential Privacy and the Decennial Census. https://assets.ipums.org/_files/intro_to_differential_privacy_IPUMS_workshop.pdf, 2019.

[38] Victoria Velkoff. Proposed 2020 Census Data Products. Presentation to JASON.

[39] James Whitehorne. Overview of redistricting data products. Presentation to JASON.

[40] Tommy Wright. Suitability Assessment of Data treated by DA Methods for Redistricting: Observations. Presentation to JASON.