Synthetic Data via Differential Privacy

ABSTRACT

Synthetic Data Generation is a popular approach to data release when the privacy of individuals in the database is a concern. This approach, used by the US Census Bureau, Department of Education, among others, gives the analyst the illusion of dealing with real data. Ideally, a synthetic data generation technique should guarantee the privacy of individuals in the database, and yet output a synthetic data set that is "representative" of the true data.

We study the problem of synthesizing data sets with the guarantee of differential privacy for the source data. Specifically, we present three differentially private algorithms for data synthesis in domains that admit hierarchical decompositions (eg: spatial data, but including other domains such as text strings). Our main technical tool (and contribution) is the private adaptive histogram, which uses non-uniform granularity to accurately resolve dense subpopulations without overfitting sparse sub-populations.

The problem of private data synthesis has been previously studied in [12], who conclude that differential privacy was too strong a guarantee to provide for accurate results. We do not find this to be the case. We evaluate our approaches on two data sets: the census commute data set used in [12], and a set of query logs released from Microsoft's Live search engine. Our experiments indicate that for large data sets, differential privacy is not at all incompatible with very accurate synthetic data.

1. INTRODUCTION

Consider a data provider such as the census bureau, that has a database, each record of which contains sensitive information about an individual. The data-provider might want to make this data available to analysts for discovering large scale patterns. However, privacy concerns require that the data be "sanitized" before release, to prevent leakage of sensitive information.

We study the problem of synthesizing representative data sets from a sensitive source data set, without compromising the privacy of the underlying records. This problem is central to privacy-preserving data publishing, where a data provider intends to publish representative information about their data so that third parties can use the data offline, commonly for exploratory data analysis, without constant interrogation of the data provider.

Our goal is to provide a general data synthesis framework, capable of targeting a broad spectrum of domains. The approaches we present in this work are aimed at data sets whose domains admit a *hierarchical decomposition*, those domains with meaningful taxonomies that recursively subdivide populations into smaller, but coherent subpopulations. A natural example are points in a d-dimensional Euclidean space: this set can be recursively partitioned into 2^d subsets, splitting each set along its midpoints in each dimension, recursively, as deeply as is needed. Note that we want the taxonomy to be fixed, independent of the data, though we will use the data to determine how far down each branch of the taxonomy we will travel.

An additional data type we consider is free text – web search queries in our case – where the taxonomy is simply defined by the letters of the alphabet (plus numbers and spaces, in our case). The domain of arbitrary text strings is refined to "those that begin with 'a'", "those that begin with 'b'", etc., with similar refinements at subsequent levels based on the character at the associated position. Clearly some domains do not admit natural or meaningful hierarchical decompositions, and we restrict our attention here to those that do.

Much recent research has been made into the formal definition of privacy, and numerous definitions have resulted. Rather than survey them, we recount the findings of [12], who conclude that, with the exception of variants of *differential privacy*, none of the privacy definitions fit the setting of synthetic data well. Our experience is that there are informal analogies to be made – the k in k-anonymity with $1/\epsilon$ from ϵ-differential privacy – but that differential privacy remains the most robust of the privacy frameworks, providing formal guarantees without imposing assumptions on prior knowledge or output structure. We will use differential privacy rather than introduce new definitions.

1.1 Related Work

Most of the prior research on data publishing has focused not on synthetic data, but rather on masking specific information in the actual records to attempt to limit the potential for re-identification. Solution concepts like k-anonymity [17], l-diversity [13], m-invariance [19], and t-closeness [11], among others [18, 5, 14] describe criteria for data release that aim to protect the source data. However, the sequence of results in this area demonstrate mainly that none of the techniques are yet sufficient; in fact Ganta et al. [8] describe and evaluate practical attacks on many such techniques. Other approaches such as [7, 16] present alternate criteria for masking data, with guarantees very much in line with differential privacy, but introduce an amount of noise into the data that grows with the size of the data.

Nevertheless, techniques with end-to-end privacy guarantees that introduce only fixed amounts of noise are possible in principle. Dwork et al. observe in [6] that the release of subpopulation counts perturbed by Laplace noise (a symmetric exponential distribution) provides differential privacy guarantees, independent of the number of subpopulations. This approach was considered by [12] for estimating densities in a census commute setting (records are pairs of source and destination census blocks), but ultimately rejected due to the severe sparsity of the data (most census source-destination pairs have zero representatives). With such sparse data sets, the noise introduced would overwhelm the signal, and the resulting data would likely be of little use. Instead, [12] use a multinomial sampling approach with a weakened definition of differential privacy, and coarsen their domain using external knowledge to enforce minimal subpopulation counts. Our approach will provide accuracy and differential privacy, without requiring auxiliary information or assumptions on the input (other than its structure).

Most prior work on differential privacy has used an *interactive* framework, in which specific questions of the data are posed and answered. In the context of differential privacy, data release is strictly less useful than interactive computation; [6] prove a separation between the accuracies that can be achieved for general queries in interactive and non-interactive settings. If an analyst has a specific question in mind, it is usually best to pose it in an interactive fashion. However, synthetic data is aimed largely at supporting exploratory data analysis, where the analyst does not yet have a fixed set of questions in mind. Moreover, synthetic data is better suited to answering a large numbers of questions with limited accuracy than an interactive interface, whose guarantees (certainly with differential privacy) tend to decay linearly with the number of questions.

Perhaps the most proximate work, theoretically, is the contingency table release of [2]. This work gives techniques, indirectly, for synthesizing data sets that respect certain measured properties of the source data (low order marginals between numerous boolean attributes). This approach uses an interactive mechanism to measure the properties privately, and then synthesize new data from these measurements, independent of the source data. This is an example of *parametric* synthetic data, where parameters of a model are estimated and used to synthesize fresh data. While very suitable if the relevant properties have been identified, such models usually reveal little in their outputs other than the values of the measured properties. In contrast, our approach is largely *non-parametric*, in that it attempts to reflect the data distribution directly, without modeling assumptions. Our hope is that it can then lead to interesting and unexpected conclusions about the source data.

Blum et al. [3] also consider differentially private data release, describing a [computationally inefficient] algorithm for synthetic data release that gives differential privacy and approximately preserves the accuracy of all concepts from a concept class of low VC dimension. They also give efficient algorithms in a few restricted cases, and an efficient algorithm for releasing sufficient information to answer halfspace queries, but in the form of aggregate statistics, rather than synthetic data.

1.2 Contributions

We present three algorithms for synthesizing data sets over domains that admit hierarchical decompositions. Our main techniques involve adaptive resolution of the domain, resolving and re-measuring regions that have substantial subpopulations and coarsening sparse regions. This allows our measurements to accurately track dense subpopulations, without overfitting sparse subpopulations (which would necessarily sacrifice one of privacy and accuracy). Our first scheme is a folklore straw-man, and is substantially improved in two different directions in the second and third schemes.

Our approaches differ from previous work in that they are explicitly designed for differential privacy, as opposed to [12] and [4] who analyze unadapted techniques (multinomial sampling from a Dirichlet prior, and random subsampling, respectively). Unlike these works, our privacy guarantees will not be a consequence of the randomness inherent in resampling, but rather enforced by algorithmic randomness in the measurement process. By side-stepping the resampling process and its inherent variance, we can substantially improve the accuracy over any resampling approach, including even those with no privacy guarantees. We provide theoretical arguments comparing the variances of our approaches, as well as empirical comparisons with other approaches.

Additionally, in our approaches the degree of privacy is a parameter to the algorithm; it can be set anywhere along a continuum from “entirely private” to “wholly disclosive”, resulting in data sets ranging, respectively, from uniformly random data to the source data itself. This is unlike the previous work of [12] and [4], whose privacy guarantees are derived from constraints on the inputs (eg: that subpopulation counts must be at least a certain value) and are not easily configured by the data synthesizer, especially if the constraints do not actually hold.

1.3 Overview of Techniques

The standard histogram measurement described in [6] observes that if a domain is *a priori* decomposed into disjoint subdomains, subpopulation counts can be released privately by adding Laplace noise to each. Machanavajjhala et al. [12] find that the sparsity of their data with respect to the size of their domain makes these techniques ineffective. Specifically, with incredibly sparse data (roughly 10^8 census respondents for over 6.4×10^{13} possible values), the contribution of actual data is dwarfed by the contribution of the noise added for privacy. Machanavajjhala et al. propose coarsening subdomains to ensure minimal cell counts, but rely on auxiliary information to provide the appropriate levels for them.

We approach the problem differently, conceptually starting from a single population and refining subpopulations whose (noisy) counts surpass some threshold (depending on the input privacy parameter). This ensures that the only subpopulations we measure are those with sufficient support as to make their measurement statistically significant. At the same time, it has the flexibility to resolve very populous subdomains and provide higher resolution information therein. This process uses only noisy measurement of counts, and provides differential privacy guarantees proportional to the depth of the recursion.

We can improve on this first cut, by observing that the decision of whether or not to further partition a region leaks less information than the noisy count itself, especially when this count is large. Implementing this intuition requires careful modifications to the algorithm and a more involved privacy analysis. We end up with a similar algorithm, but one whose privacy cost is independent of the depth of recursion.

We also present an approach for direct data synthesis following the same spirit as the first approach, but rather than have the noisy counts determine whether to refine or not, we have them commit to a subpopulation count, which the subsequent recursive steps must allocate. This has the appealing property that macroscopic properties are preserved with a great deal of accuracy; the total number of synthesized data points are within additive constant error of the true value, and the same statement roughly holds for all subpopulations. For sizeable subpopulations, this approach can be substantially more accurate even than resampling from the distribution producing the true data, whose variance is linear in the subpopulation size, rather than constant.

Our experiments indicate that our techniques are somewhat incomparable. Although both the second and third approach improve on the first, straw-man approach, they do so in different ways. The improved adaptive histogram approach does well at identifying structure to a high level of accuracy, producing good maps, and lists of search queries. The direct synthesis approach, on the other hand, has a much easier time of producing data sets that respect large scale statistics without introducing noticeable bias.

1.4 Paper Outline

We start in Section 2 with an introduction to notation, the definition of differential privacy, and the development of a basic adaptive histogram scheme. In Section 3 we improve on this scheme, presenting a slight modification and detailed privacy analysis. In Section 4 we present a related approach that bypasses the histogram step, and synthesizes data directly. We measure and evaluate the performance of these three approaches in Section 5 on two different data sets. Finally, we conclude in Section 6 with closing remarks and future directions for research. Appendix A contains an extended discussion of alternate implementations of one of our approaches, and Appendix B contains several anecdotal examples of data we are able to synthesize, both for census commuters as well as actual web searches. Several code examples follow the appendices.

2. TECHNICAL PRELIMINARIES

We start by introducing some notation. Rather than think of the domain D as partitioned *a priori* into a large number of small parts, we imagine a hierarchy of partitions, so that D is first split into a few parts not much smaller than D. These parts are then further partitioned into a few slightly smaller parts, and so on. Rather than think of records as elements of D, we will imagine them as sequences of a different type K that defines which subpopulation they fall in each of the refinement steps.

As a concrete simple example, suppose each record describes a point on the unit interval $[0, 1]$. We can describe each point by its binary representation, which is a sequence of 0s and 1s. This corresponds to recursively decomposing the interval into two subintervals of half the length, and the binary representation simply indicates the sequence of subintervals the point lies in. We will often consider spatial data sets of geometric points from the d-dimensional square: $[0, 1]^d$. We can now transform each point into a sequence of characters from the set $\{0, 1\}^d$, each bit indicating the next bit in the fractional component of the corresponding coordinate. Geometrically, we have decomposed the d-dimensional square into 2^d sub squares, each of half the radius, and the sequence of bit patterns indicates the sequence of subsquares the data point lies in. Text strings are another example, where we simply take K to be the alpha-numeric characters and use the characters at each position to drive the refinement.

2.1 Differential Privacy

Differential privacy is a property of randomized computations that ensures that a single change to the input of the computation results in a limited *relative* change to the probability of any output or sets of outputs.

Definition 1. *We say a randomized computation M provides* ϵ-differential privacy *if for all data sets A and B and all $S \subseteq Range(M)$,*

$$\mathbf{Pr}[M(A) \in S] \quad \leq \quad \exp(\epsilon|A \Delta B|) \times \mathbf{Pr}[M(B) \in S] \ , \ (1)$$

where $A \Delta B$ denotes the symmetric difference of A and B.

Intuitively, the presence or absence of any single record is not substantially reflected in the output distribution. This assures participants whose records constitute the dataset that the presence or absence of their data from the data set will not result in noticeably different behavior of those who observe the result of the computation.

The easiest example of a differentially private computation, and one we will use extensively, is the “noisy count”, in which the correct count of the number of records in a data set is perturbed by a Laplace random variable with parameter $1/\epsilon$. The Laplace distribution is a symmetric exponential distribution with density $\epsilon \exp(-\epsilon|x|)/2$, and provides ϵ-differential privacy despite its exponential tails in both directions giving substantial [though imperfect] accuracy. For completeness, we reproduce the proof from [6].

Let A and B be two datasets with symmetric distance 1. Thus the counts $c(A)$ and $c(B)$ differ by at most 1. The probability density at x for input A is

$$\begin{aligned} & \epsilon \exp(-\epsilon|x - c(A)|)/2 \\ \leq\ & \epsilon \exp(-\epsilon(|x - c(B)| - |c(A) - c(B)|)/2 \\ \leq\ & \exp(\epsilon) \cdot \epsilon \exp(-\epsilon|x - c(B)|)/2, \end{aligned}$$

where we have used the triangle inequality in the first step.

2.2 Static Histograms

Dwork et al [6] observed that for any partitioning of the input domain into any number of parts, releasing the subpopulation counts with Laplace (a symmetric exponential distribution) noise of parameter $1/\epsilon$ added to each gives ϵ-differential privacy. This was a substantial improvement on previous analyses, which suggest that the number of parts of the partition would need to appear in the noise term, or the denominator of the privacy guarantee, if the measurements were to be taken independently. This is not so, [6] argues, because a single participant cannot arbitrarily influence each count; any individual participates in at most one count.

Remark: The previous work of [12] can be viewed as analogous to static histograms. In that work, the data (source-destination pairs) are pre-partitioned by destination, which they treat as a public attribute, and for each a histogram over sources is produced. Their main concern is the sparsity of the data: there are roughly eight million census blocks and only tens to hundreds of commuters arriving at each. Consequently, simple noisy measurement would make eight million noisy measurements for each destination block, and as a result introduce a tremendous number of spurious and unrepresentative records.

2.3 Adaptive Histograms

It is possible to [carefully] extend the analysis from [6] from counts of disjoint populations, to general differentially private computation on disjoint subpopulations. Doing so requires the alternate definition we use (based on symmetric difference) rather than indistinguishability [6] (based on hamming distance).

Theorem 1. *Let $\{D_i\}$ be disjoint subsets of D, and let M provide ϵ-differential privacy. Partitioning an input set X by the D_i and executing M on each, ie the set: $\{M(X \cap D_i)\}$, provides ϵ-differential privacy.*

Proof. As in [6], using the definition of differential privacy instead of that of the Laplace distribution. □

This generalization lets us design a very simple adaptive histogram approach: if a data set has size at least a given threshold (measured via a differentially private noisy count), we subdivide the set and recursively apply the algorithm on each of the subparts. Figure 1 gives the pseudocode for the algorithm. We will run the algorithm for at most B levels of the recursion, for a suitable parameter B.

```
Algorithm Basic(L, thresh)
If (L.Noisycount($\epsilon$) $\geq$ thresh ) then
    Partition L into parts L$_1$, ..., L$_k$.
    For $i = 1, \ldots, k$, set $H_i$ = Basic($L_i$,thresh).
    return the composition of the $H_i$'s.
Else return (L,L.Noisycount($\epsilon$)).
```

Figure 1: Basic Adaptive Histogram

Corollary 2. *The basic adaptive histogram algorithm from Figure 1, run to depth at most B, provides (ϵB)-differential privacy.*

Proof. : The recursive call at depth i provides differential privacy equal to that of the noisy count, ϵ, plus the differential privacy of the executions at depth $i + 1$. Since we stop at depth B, the claim follows by induction. □

This adaptive approach lets us drill down into dense subpopulations, while keeping a coarse approximation to sparse subpopulations. The set of prefixes that result provide an interesting view of the data themselves, but we will use them simply to re-measure the data. The set of prefixes structurally partition the input domain, and we run a second [static] histogram query using this partition to measure the counts. For each part, we synthesize a number of representatives equal to this count.

Remark: We run this second pass, rather than use the final, terminal counts, for two reasons: Firstly, we may want to use higher accuracy in the final measurement than in each step of the recursive descent, and secondly the final counts, by nature of being final, are negatively biased; recounting gives us unbiased estimators for the counts. The privacy cost of the data synthesis increases from that of adaptive histogramming by the reciprocal of the accuracy of this second pass.

3. IMPROVED ADAPTIVE HISTOGRAMS

Our improved adaptive histogram is nearly identical to the naive algorithm, with two important changes. We will no longer check the depth of the computation, but we continue the recursive computation on a strict subset of the input data set, discarding a small set of records from each recursive call to ensure that the counts strictly decrease by a sufficient margin.

The resulting algorithm appears in Figure 2. Note that this is identical to the naive computation, except for the invocation of `Skip(discount)`, removing `discount` records before each recursive call. This apparently minor change, and a more thorough analysis of differential privacy, lead to privacy bounds independent of the depth of the recursion. We note that the `Skip` subroutine is underspecified; in this section, we just assume it removes the first `discount` records from the data set. We defer a discussion on more intelligent choices of records to remove, and their implications, to Appendix A.

```
Algorithm Improved(L, thresh)
If (L.Noisycount($\epsilon$) $\geq$ thresh ) then
    L.Skip(discount).
    Partition L into parts L$_1$, ... ,L$_k$.
    For $i = 1, \ldots, k$, set $H_i$ = Improved($L_i$,thresh).
    return the composition of the $H_i$'s.
Else return (L,L.Noisycount($\epsilon$)).
```

Figure 2: Improved Adaptive Histogram

Theorem 3. *: The adaptive histogram approach from Figure 2, when executed with* `discount` $= t \geq b + \ln 2/\epsilon$, *provides* (2ϵ)*-differential privacy, where* $b =$ `threshold` *is the termination threshold.*

Proof. The probability associated with any possible sequence of outputs is the product of the probabilities of continuation at each internal node in the computation tree and the product of termination probabilities at each leaf. Let $p(x)$ be the probability of continuation given a true count of x. The addition of Laplace noise implies that

$$p(x) \quad = \quad \begin{cases} 1 - \exp(-\epsilon|x-b|)/2 & \text{if } x \geq b \\ \exp(-\epsilon|x-b|)/2 & \text{if } x < b \end{cases} \qquad (2)$$

This observation allows us to work only with the true counts, instead of the noisy counts. If A and B differ on a single record, the true counts can only differ along a single path down the computation tree. In a simple world, this would simply be along the path containing their difference, whose counts differ by one. In fact, the record in difference could be one that is skipped, causing no further change in the counts along its path, but resuscitating a different record that increases the counts along its path. It in turn may then be skipped, and so on, causing a chain reaction that, fortunately, follows a simple path and never grows beyond one record in difference. The counts in this path may become zero, and thus equal, but we continue the path to the leaf of the tree nonetheless, observing that these probabilities will eventually cancel.

As the counts along the computation tree are equal everywhere except a path P, the ratio of continuation and termination probabilities will nicely cancel everywhere except along that path. Letting a_i and b_i be the counts of A and B, respectively, at the ith node on this path from the leaf,

$$\frac{\mathbf{Pr}[M(A) \text{ contains } P]}{\mathbf{Pr}[M(B) \text{ contains } P]} \quad = \quad \prod_{i>0} \frac{p(a_i)}{p(b_i)} \times \frac{1-p(a_0)}{1-p(b_0)} \; . \; (3)$$

If an element is removed from B yielding A, each term $\frac{p(a_i)}{p(b_i)}$ is at most one and the final term is at most $\exp(\epsilon)$. If an element is added to B yielding A, the final term is at most one, but the terms in the product can each be greater than one and must be bounded.

Our plan is to take logarithms, changing the product of probabilities to a summation of its logarithms. To start out, we observe that the ratio of $p(x+1)/p(x)$ is greatest when x is smallest, and decreases monotonically as x grows, and that *log* is a monotone function. We can thus upper bound the logarithm of the ratio by the average over the preceding interval of size t,

$$\log\left(\frac{p(a_i)}{p(b_i)}\right) \quad \leq \quad \frac{1}{t} \sum_{x=b_i-t+1}^{b_i} \log\left(\frac{p(x+1)}{p(x+0)}\right) \; . \qquad (4)$$

The decrease of true counts by at least t ensures that these sums are disjoint. Since $x \geq 0$ for every node where the counts differ, we can bound

$$\sum_{i>0} \log\left(\frac{p(a_i)}{p(b_i)}\right) \quad \leq \quad \frac{1}{t} \sum_{x=-t}^{\infty} \log\left(\frac{p(x+1)}{p(x+0)}\right) \; . \qquad (5)$$

This summation, using the observation that the logarithm of ratios is just a difference of logarithms, telescopes, giving

$$\frac{1}{t} \sum_{x=-t}^{\infty} (\log p(x+1) - \log p(x+0)) \quad \leq \quad \frac{1}{t} \log\left(\frac{1}{p(-t)}\right) (6)$$

Rewriting this bound in the product formulation, we get

$$\prod_{i>0} \frac{p(a_i)}{p(b_i)} \quad \leq \quad \exp(\epsilon(t+b)/t) \times 2^{1/t} \; . \qquad (7)$$

Recalling that $t = b + \ln 2/\epsilon$, and simplifying, we can bound the right hand side by $\exp(2\epsilon)$. □

Remark: In general, we should take b to be at least the natural log of the branching factor – the number of subpopulations of any given population – divided by epsilon. Otherwise the computation has non-zero probability of diverging, producing an infinite histogram (note: privacy is guaranteed either way).

In appendix A, we discuss alternative discounting functions that can replace the `Skip` operation above, that may be more suited to preserve different structural properties of the data, while continuing to guarantee privacy.

4. DIRECT DATA SYNTHESIS

Our approach to direct data synthesis uses the same principles as the basic adaptive histogram approaches. However, rather than recursively subdivide and recount each subpopulation, our data synthesis commits each population to a fixed count, and then subdivides this count among its subpopulations. This approach ensures that macroscopic properties stay accurate, independent of the ultimate resolution of the data. Unlike parametric "model and resample" approaches, with count variances linear in the counts, we will see that our variances depends only on the accuracy with which we take our noisy counts. Although this can be larger for some populations (notably "structural zeros", where data is known not to exist), for many populations of interest it is not.

Algorithmically, our approach takes as input a parameter `maxdepth`, and a number of records to synthesize, perhaps determined using a noisy count over the whole population. It then allocates exactly this number of slots to its subpopulations, privately, using noisy subpopulation counts. Each

```
Algorithm Synthesize($L$, count, prefix, maxdepth)
If (count == 0) return
If (count == 1) output prefix
If (maxdepth == 0) output count copies of prefix
Partition $L$ according to key to get $L_1, \ldots, L_k$
subcount$_i$ ← Noisycount($L_i$)
Invoke Center(count, subcounts).
For $i = 1, \ldots, k$,
    Synthesize($L_i$,subcount$_i$,prefix+'i',maxdepth-1)
```

Figure 3: Direct Data Synthesis

```
Algorithm Center(count, subcounts)
For $i = 1, \ldots, k$, set subcount$_i$ ← max(0,subcount$_i$).
direction ← sgn(count − $\sum_i$ subcount$_i$)
Repeat in roundrobin order
    subcount$_i$ ← max(0,subcount$_i$+direction)
until (count == $\sum_i$ subcount$_i$)
```

Figure 4: Count Centering for Direct Data Synthesis

subpopulation recursively allocates their allocations, and so on. The recursion continues as long as the count to allocate is greater than one, or if the depth exceeds `maxdepth`, and terminates otherwise. The allocation of the slots defines a data set, with one record produced from each slot. Figure 3 demonstrates the process.

There are many ways to allocate counts based on noisy subpopulation counts, and care needs to be taken with issues such as bias and round-off. In Figure 4 we choose the primitive but effective approach of rotating through each of the subpopulations, moving the associated subpopulation count one unit in the correct direction, if possible, until their sum equals the intended count. This approach has the appealing property that the subpopulation error, the difference between each subpopulation count and the value it is allocated (and consequently the count observed in the output), is only the error in measurement, plus an even share of the error introduced between the allocated count and noisy subcount measurements. Consequently, the error from previous recursive levels typically dissipates, and each subpopulation enjoys the same level of accuracy.

More precisely, suppose that each noisy count is computed by adding Laplace noise with standard deviation R. Then the count at each node gets an expected error of about R, plus the error inherited from the parent during the centering procedure. A simple calculation shows that for a node at depth d, the expected error in the count is no larger that dR. This bound is close to tight in the worst case (when the data is all zeroes except for one node at depth d) and we cannot hope to prove a better bound without making additional assumptions on the data. However, in most data sets the expected error from the parent would be distributed equally across several children, and hence the actual error in count at a node at depth d is $O(R)$ independent of the nodes depth. Indeed our experiments indicate that this technique is comparable to the adaptive histogram technique on real data sets. Finally, note that R itself must be `maxdepth`$/\epsilon$ to get ϵ-differential privacy.

4.1 Connections to k-Anonymity

Our data synthesis approach continues to allocate counts until a single record remains, but realistically we lose signal once the count drops below $1/\epsilon$. If we were inclined, we could simply publish the prefix at this point as well as the final noisy count for the population (perhaps taken with higher accuracy, as one would do in a histogram). At this point, the approach begins to resemble the kin of k-anonymity, which group the records of the data set so that each group contains at least k records (or various other, more demanding properties, to strengthen the protection).

One important difference is that k-anonymization approaches are free to choose from many generalizations of a set of records; they are not required to pick a prefix of the description of the record, and indeed the records usually come from a multi-attribute lattice, rather than a single fixed hierarchy. Each step of refinement is responsible not only for determining which records go where, but further which attribute is used to make the decision. Another important difference is that k-anonymity does not provide formal end-to-end privacy guarantees.

Fortunately, we can accommodate this added functionality in the differential privacy setting using the exponential mechanism of [15]. The exponential mechanisms provides a differentially private mechanism for choosing among several discrete outcomes (here, which attribute to refine next) based on a score function that must satisfy certain stability properties (that a single record not change the score by more than one). One natural choice is to use "imbalance" of an attribute as the score function, hoping to pick those attributes that partition the data as little as possible:

$$\text{Imb}(\text{attribute } A, D) \quad = \quad \max_{a_i \in A} |x \in D : A(x) = a_i| \qquad (8)$$

This score function leads to a distribution that selects attribute A with probability proportional to $\exp(\epsilon \times \text{Imb}(A, D))$. This approach is purely heuristic; many other algorithms give formal approximation guarantees with respect to optimal k-anonymization which we do not claim. Nonetheless, it is an approach to k-anonymization that provides the guarantees of differential privacy.

5. EXPERIMENTAL VALIDATION

We now consider two real data sets drawn from different domain: a data set of commute patterns derived from the Longitudinal Employer-Household Dynamics Survey, and a data set of search queries issued to Microsoft's Live Search. For each of the data sets, we will consider our three approaches: basic adaptive histograms, improved histograms, and direct synthesis, with three settings of differential privacy: 10.0, 1.0, and 0.1. Because of the difference in domains, spatial and non, we will use different metrics and benchmarks in the two settings, described in more detail in their respective subsections.

5.1 Census Commute Data

While the Longitudinal Employer-Household Dynamics data set (LEHD) contains a substantial amount of data, we are specifically interested in the commute records, which have for each respondent the census blocks of employment and residence. This is the data set considered by [12], and synthetic data based on this data set is currently in deployment. In fact, the only data we have access to are themselves synthetic, but like [12], we will treat them as ground truth. The census bureau actually releases three different synthetic data sets; we use one of them as the ground truth, and the others as benchmarks for accuracy.

We start by transforming pairs of census blocks to two pairs of latitude-longitude coordinates based on the centroids of the census blocks. We then normalize these to the unit four dimensional cube. The decomposition we use will be the four dimensional analogue of a quadtree, decomposing such cubes into sixteen sub-cubes of half the radius. As mentioned, we compare three different techniques, with three different privacy settings, but we have additional data in the form of parallel census releases.

We assess our synthetic data using several different metrics. We first consider in Figure 5 an approximation to the earth-mover distance between two data sets, provided by Indyk [9], using the sum across all subpopulations, at all scales, of the diameter of the subpopulation times the symmetric difference of the two sets on it.

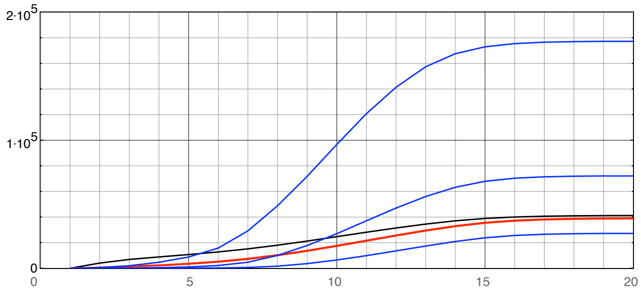


Figure 5: Measurement of the earthmover approximation. The y axis measures the aggregate error, up to histogram depths indicated by the x axis. The lines plotted are: in blue, three direct synthesis approaches, with privacy values 0.1, 1.0, and 10.0, in red two alternate census releases, and in black the expected distance for a random resampling of the source data (with cell variance proportional to its count).

We take several conclusions from these measurements. First, the direct data synthesis (blue lines) *can* outperform even the alternate census releases, drawn from the same distribution as the source data. Second, the alternate census releases (red lines) do not appear to track the anticipated variance for the cell counts (black line), suggesting that they may use a more intelligent hierarchical decomposition, which we could also take advantage of with more investment in domain knowledge. Finally, as privacy increases, accuracy decreases, and we see smooth transitions suggesting that configuring the techniques to meet specified privacy or accuracy goals should not be difficult.

We also consider in Figures 6 and 7, as done in [12], the distributions of commute distances from several source locations. For figure 6 we use Honolulu, HI, and San Francisco, CA as the sources. We note that despite drawing four lines for each (three privacy settings, and the correct distribution), all four effectively occlude each other.

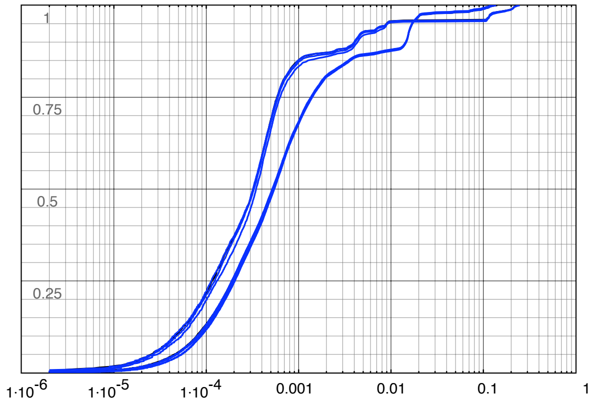


Figure 6: Cumulative density functions for commute distances for Honolulu, HI and San Francisco, CA. The true distribution is in black, with synthetic data from improved adaptive histograms with privacy settings 10.0, 1.0, and 0.1 in blue.

For a less appealing example, Figure 7 considers Billings, MO. Here we can clearly see a disconnect between the true distribution (in black) and the synthesized distributions (in blue). None are especially appealing, possibly a consequence of Montana being insufficiently resolved to accurately isolate Billings' distribution from Montana's generally.

Finally, in Table 1 in the appendix we present purely anecdotal visual evidence, plotting the distribution of commute destinations for fixed origins and comparing with ground truth. The points in these pictures have been moved to the centroid of the nearest actual census block for aesthetic reasons. This is strictly a post-processing operation that does not use the protected commute data.

5.2 Search Engine Queries

Microsoft has released a set of search queries (among other data) to external researchers as part of its "Beyond Search" request for proposals. We transform the text of each search into a sequence over the the 26 possible letters, 10 possible numerals, a period, and a blank space. Strings containing other characters are discarded, leaving us with approximately 2/3 of the original data, approximately 90 million queries.

Rather than measure the earthmover distance, which fits text badly (errors in the final few characters are as distract-

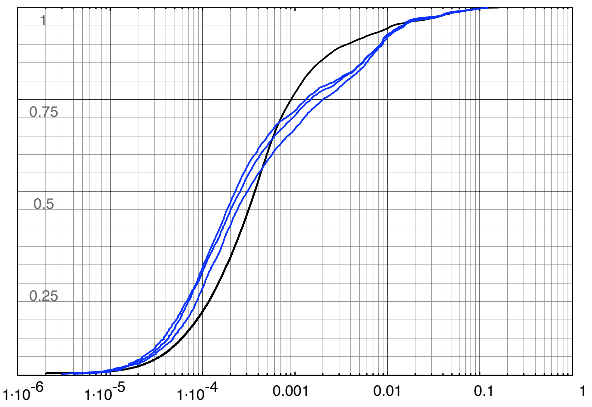


Figure 7: Cumulative density functions for commute distances for Billings, Montana. The true distribution is in black, with synthetic data from improved histograms with privacy settings 10.0, 1.0, and 0.1 in blue.

ing as in the first few characters), we will assess the fraction of the corpus that we can recover privately. That is, we will synthesize a set of words, and measure the number of actual queries that can be found in the synthetic data set, counting multiplicities. Ideally, we should be able to reconstruct a large fraction of those strings that appear with sufficient frequency, while missing the bulk of the strings that appear infrequently (as necessitated by differential privacy).

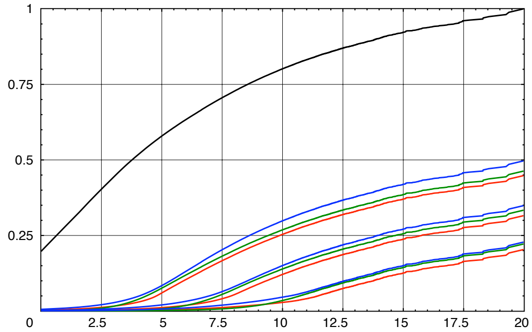


Figure 8: The cumulative fraction of queries from the source data that can be found in the synthetic data set (on the y-axis), with the x-axis being the log of the frequency of the terms. The lines are first the data cdf itself (in black), then three bundles of techniques (improved histograms, direct synthesis, and naive histograms, in that order), for privacy settings 10.0, 1.0, and 0.1.

Figure 8 and Figure 9 contain the relevant measurements. In Figure 8 we plot the cumulative distribution of recovered queries, accumulated by the frequency of the query. The various curves have the most disparity early in this aggregation (disagreeing on low frequency queries) and so we also plot the same data in Figure 9, reversing the order of integration: from most frequent to least frequent. The second figure demonstrates the high fidelity on frequent queries more clearly.

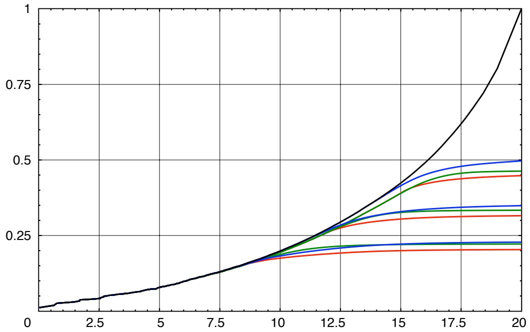


Figure 9: The same data as Figure 8, with the order of integration reversed: the x-axis is 20 minus the log of the frequency.

The key observations from the figures are that, as expected, all techniques have relatively high fidelity on high frequency terms. The point at which the three techniques break down is determined more by the privacy setting (10.0, 1.0, and 0.1 each corresponding to a bundle of three lines in the figures) than by the technique. One minor, but interesting detail can be seen in Figure 9, which is that the best of the three techniques (improved histograms) is tight to the ground truth for noticably longer than the other techniques. This is a consequence of the fixed depth of recursion (20, in our experiments) the other approaches have built in. As the adaptive histograms can descend to arbitrary levels, they are able to capture this additional fraction of long and frequent queries.

6. CONCLUSIONS / FUTURE DIRECTIONS

We have presented three techniques for synthesizing synthetic data sets with the guarantees of differential privacy. The techniques use adaptive resolution histograms to track the local density of the data sets, producing representative samples at multiple scales. We evaluated these techniques on two modern and complex data sets, demonstrating the feasibility of accurate and private data release.

Our approaches are very simple, and have the potential to be specialized in several directions. In many settings, valuable domain knowledge exists about, e.g., commute patterns or the english language. Incorporating this information into the synthesis routines, or in post process, can lead to substantially more realistic and accurate data sets.

It is also interesting to investigate whether our approach can be unified with other more parametric approaches. For example, the contingency release of [2] apply very well for data sets of numerous boolean variables, whereas our approaches apply best to data sets of few, but complex attributes. Bringing these two approaches together in a common framework will be necessary for the richest of data sets.

Acknowledgments

7. REFERENCES

[1] *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, April 15-20, 2007, The Marmara Hotel, Istanbul, Turkey*. IEEE, 2007.

[2] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In L. Libkin, editor, *PODS*, pages 273–282. ACM, 2007.

[3] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In R. E. Ladner and C. Dwork, editors, *STOC*, pages 609–618. ACM, 2008.

[4] K. Chaudhuri and N. Mishra. When random sampling preserves privacy. In C. Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 198–213. Springer, 2006.

[5] B.-C. Chen, R. Ramakrishnan, and K. LeFevre. Privacy skyline: Privacy with multidimensional adversarial knowledge. In Koch et al. [10], pages 770–781.

[6] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, editors, *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

[7] A. V. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS*, pages 211–222. ACM, 2003.

[8] S. R. Ganta, S. Kasiviswanathan, and A. Smith. Composition attacks and auxiliary information in data privacy. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining (SIG-KDD)*, 2008.

[9] P. Indyk. Algorithms for dynamic geometric problems over data streams. In L. Babai, editor, *STOC*, pages 373–380. ACM, 2004.

[10] C. Koch, J. Gehrke, M. N. Garofalakis, D. Srivastava, K. Aberer, A. Deshpande, D. Florescu, C. Y. Chan, V. Ganti, C.-C. Kanne, W. Klas, and E. J. Neuhold, editors. *Proceedings of the 33rd International Conference on Very Large Data Bases, University of Vienna, Austria, September 23-27, 2007*. ACM, 2007.

[11] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE* [1], pages 106–115.

[12] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *In Proc. ICDE*, 2008.

[13] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *TKDD*, 1(1), 2007.

[14] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern. Worst-case background knowledge for privacy-preserving data publishing. In *ICDE* [1], pages 126–135.

[15] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103. IEEE Computer Society, 2007.

[16] V. Rastogi, S. Hong, and D. Suciu. The boundary between privacy and utility in data publishing. In Koch et al. [10], pages 531–542.

[17] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *PODS*, page 188. ACM Press, 1998.

[18] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. In U. Dayal, K.-Y. Whang, D. B. Lomet, G. Alonso, G. M. Lohman, M. L. Kersten, S. K. Cha, and Y.-K. Kim, editors, *VLDB*, pages 139–150. ACM, 2006.

[19] X. Xiao and Y. Tao. M-invariance: towards privacy preserving re-publication of dynamic datasets. In C. Y. Chan, B. C. Ooi, and A. Zhou, editors, *SIGMOD Conference*, pages 689–700. ACM, 2007.

APPENDIX

A. GENERALIZED DISCOUNTING

The techniques we applied for improved adaptive histograms can be generalized substantially, replacing the primitive `Skip` operation with more saavy discounting of counts. To articulate the more general setting, it will be important to consider the possibility of incompletely determined records: those whose prefix does not carry them to a leaf in the taxonomy, but deposits them instead at an internal node. We implicitly extend the domain D to include such partial records, if it does not already contain them.

We call a function $T : D^n \to D^n$ a t-discount function if its outputs ensure the property that for each population, each of its subpopulation has at least t fewer records. Importantly, we will also require the function to be *stable*, in the formal sense that for any inputs A and B,

$$|T(A)\Delta T(B)| \quad \leq \quad |A\Delta B| \; . \qquad (9)$$

The main theorem of Section 3 holds with `Skip` replaced by any stable t-discount function.

There are several stable t-discount functions that may prove interesting in shaping the output histograms to reflect different measurement goals. Examples include:

- **Uniform Discounting**: We can remove an equal number of records from each subpopulation. Cases when the some subpopulations are empty (or too small), or t is not divisible by the number of subpopulations, can and should be handled carefully to ensure stability.

- **Proportional Discounting**: Like Skip, we could simply truncate each record to the first subpopulation containing it which does not yet have t representatives. Randomly permuting the input before applying this technique gives a form of proportional discounting, where subpopulations are discounted in proportion to their population.
- **Progressive Discounting**: We can remove the records from the most populous subpopulation, so long as we make sure to share the burden among other populations with large population. Repeatedly removing a single record from the currently largest subpopulation has this property. Note that the same approach with the smallest subpopulation is not stable.
- **Fractional Discounting**: All of these techniques, and the associated proofs, hold when records are not removed so much as *diminished* by a fractional amount. The aggregate weight of a subpopulation still must decrease by t, but we do not require integral weights.

Each of these discounting schemes results in qualitatively different output histograms, focusing on different interesting aspects of the data. The uniform/proportional/progressive discounting schemes range from deep-but-narrow to broad-but-shallow histograms.

Remark: Rather than explicitly transform the data, most transformations are more efficiently implemented *implicitly*, as discounts folded directly into the subpopulation's noised counts, without explicitly identifying the record to be removed. This efficiently permits the substantial flexibility of deferring the decision as to which sub-subpopulation counts we wish to discount until we visit it.

B. SYNTHETIC DATA EXAMPLES

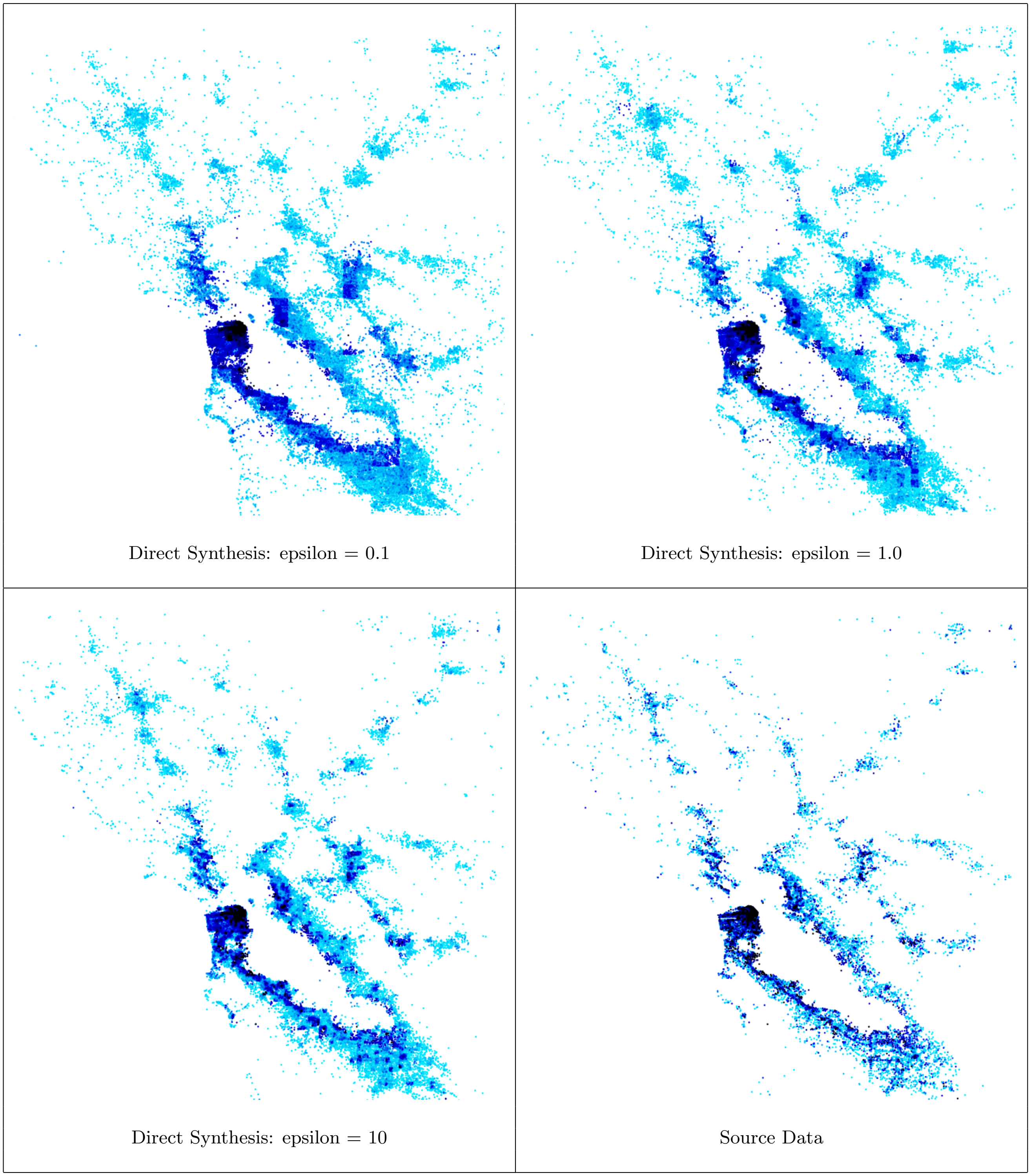
In this section we present a few examples of the synthesized data, to get a feel for what it looks like and to what degree the data feels authentic.

In Figure 10 we present a few of the synthetic queries produced using one of our approaches: the improved adaptive histograms with epsilon set to 1.0. We present an arbitrary subset of the queries, restricted to those that appear at least 500 times. This choice is mostly aesthetic; lower thresholds produce lists that show less diversity (hundreds of queries prefixed with 'abc' rather than ten or so) and higher thresholds result in unsurprising results. The most interesting feature, from our point of view, is that the synthesized queries are actual identifiable searches we might expect to see. This trend holds across the synthetic data set.

Table 1 contains views of three synthetic data sets, using the direct data synthesis approach with parameters 0.1, 1.0, and 10.0. We draw heatmaps for commute destinations originating in San Francisco, CA, but stress that the data was not synthesized with this subpopulation in mind. The fidelity clearly improves as the parameters increase, but always remains more diffuse than the source data.

```
  548 aa route finder
  724 aa route planner
 7291 aa.com
 5256 aaa
  742 aaa travel
 2146 aaa.com
 1362 aafes
  538 aafes.com
 1021 aapl
 3853 aarp
  595 abby winters
11557 abc
 1105 abc dancing with the stars
 1127 abc daytime
 2623 abc distributing
  666 abc family
  832 abc good morning ameri
 5474 abc news
 1974 abc tv
  548 abc tv shows
11632 abc.com
  683 abc13
  560 abc13.com
  915 abc7
  960 abc7.com
  556 abcdistributing
  682 abcdistributing.com
  902 abcnews
 2103 abcnews.com
 2648 abercrombie
 1482 abercrombie and fitch
  741 abortion
 1233 about.com
 1404 abv.bg
  519 ac ir shiraz
  824 ac moore
 1230 academy
 1105 academy sports
 1921 access hollywood
 6737 access my aol mail
 1225 accu weather
  601 accurint
 6485 accuweather
 2240 accuweather.com
  886 ace
 2359 ace hardware
  812 ace.com
 1277 acer
  551 acid reflux
  567 acrobat
 1193 acrobat reader
  975 across the universe
  621 acs
 1390 act
  668 active x
  571 activesync
  650 activex
 1368 acura
  740 ad aware
```

Figure 10: Some synthetic queries and counts using Improved Adaptive Histograms with epsilon = 1.0.



Direct Synthesis: epsilon = 0.1

Direct Synthesis: epsilon = 1.0

Direct Synthesis: epsilon = 10

Source Data

Table 1: Heatmaps of the destinations of synthetic workers originating within 5 miles of San Francisco.

```
    // Recursively partition input until at most threshold elements remain
    public void Basic(PINQueryable<string> input, string prefix)
    {
        if (input.NosiyCount(epsilon) < threshold)
            Console.WriteLine(prefix);

        else foreach (var part in input.Partition(keys, x => x[0]))
            Basic(part.Value.Select(x => x.Substring(1)), prefix + part.Key);
    }
```

Figure 11: A basic adaptive histogram: The input data set is recursively partitioned as long as the size of the sub-population is sufficiently large.

```
    public void Histogram(PINQueryable<string> input, string prefix)
    {
        if (input.NoisyCount(epsilon) < threshold)
            Console.WriteLine(prefix);

        else foreach (var part in input.Skip(discount).Partition(keys, x => x[0])
            Histogram(part.Value.Select(x => x.Substring(1)), prefix + part.Key);
    }
```

Figure 12: An improved adaptive histogram: Before each recursive call a fixed number `discount` **of records are removed from the data set.**

```
// allocates count between children, continuing until count is at most one.
public void Synthesize(PINQueryable<string> input, string prefix, int count)
{
  if (count == 1)  Console.WriteLine(prefix);
  if (count <= 1)  return;

  var parts = input.Partition(keys, x => x.Substring(0,1));
  var subcounts = keys.Select(key => new Pair<string,int>(key,parts[key].NoisyCount(epsilon)));

  /* center subcounts so that 0 <= subcount[i] <= count and subcount.Sum() == count */

  foreach (var key in keys)
    Synthesize(parts[key].Select(x => x.Substring(1)), prefix + key, subcounts[key]);
}
```

Figure 13: Direct data synthesis skeleton. A count is allocated to a population, which then uses noisy subcounts to allocate exactly that many records to its subpopulations.

```
  /* code fragment used to center the subcounts. */

  foreach (var key in keys)
    subcounts[key] = Math.Min(count, Math.Max(0, subcounts[key]));

  int subcount = keys.Select(key => subcount[key]).Sum();
  int direction = (count > subcount) ? +1 : -1;

  for (int i = System.Random(); count != subcount; i++)
  {
    if ((direction == +1 && subcounts[keys[i % keys.Length]] < count) ||
        (direction == -1 && subcounts[keys[i % keys.Length]] > 0))
    {
      subcounts[keys[i % keys.Length]] += direction;
      subcount += direction;
    }
  }

  /* code fragment ends; subcounts are centered. */
```

Figure 14: Centering noisy subpopulation counts so that they provide the correct total.