Bioinformatics, 37, 2021, i151–i160
doi: 10.1093/bioinformatics/btab292
ISMB/ECCB 2021

Expected 10-anonymity of HyperLogLog sketches for federated queries of clinical data repositories

Ziye Tao[1], Griffin M. Weber[2] and Yun William Yu[1,3,*]

[1]Department of Mathematics, University of Toronto, Toronto, ON M5S 1A1, Canada, [2]Department of Biomedical Informatics, Harvard Medical School, Boston, MA 02115, USA and [3]Computer and Mathematical Sciences, University of Toronto at Scarborough, Scarborough, ON M1C 1A4, Canada

*To whom correspondence should be addressed.

Abstract

Motivation: The rapid growth in of electronic medical records provide immense potential to researchers, but are often silo-ed at separate hospitals. As a result, federated networks have arisen, which allow simultaneously querying medical databases at a group of connected institutions. The most basic such query is the aggregate count—e.g. How many patients have diabetes? However, depending on the protocol used to estimate that total, there is always a tradeoff in the accuracy of the estimate against the risk of leaking confidential data. Prior work has shown that it is possible to empirically control that tradeoff by using the HyperLogLog (HLL) probabilistic sketch.

Results: In this article, we prove complementary theoretical bounds on the *k*-anonymity privacy risk of using HLL sketches, as well as exhibit code to efficiently compute those bounds.

Availability and implementation: https://github.com/tzyRachel/K-anonymity-Expectation.

Contact: ywyu@math.toronto.edu

1 Introduction

Clinical data containing patients' personal medical records are important resources for biomedical research. Fully centralizing that data may permit the widest array of potential analyses, this is often not feasible due to privacy and confidentiality requirements (Benitez and Malin, 2010; Emam *et al.*, 2009; Heatherly *et al.*, 2013). During times of pressing need, such as during a global pandemic, these privacy requirements may be justifiably relaxed (Haendel *et al.*, 2020)—such as using trusted third party vendors such as Datavant (Kho and Goel, 2019)—but even then, it is important to keep in mind the various privacy-utility tradeoffs (Bengio *et al.*, 2021, 2020). A more privacy-friendly alternative is to use a federated network instead, which give hospitals control over their local databases; then, a distributed query tool enables researchers to send queries to the network, such as 'how many patients across the network have diabetes' (Brat *et al.*, 2020; Weber, 2015). A number of these hospital networks have emerged, including the Shared Health Research Information Network for Harvard affiliated hospitals (Weber *et al.*, 2009), the Federated Aggregate Cohort Estimator developed through a collaboration of five universities and institutions (Wyatt *et al.*, 2014), the open-source PopMedNet (Davies *et al.*, 2016) and the Patient Centered Outcomes Research Institute launched PCORnet as a network of networks (Fleurence *et al.*, 2014).

However, patients often receive medical care from multiple hospitals, so medical records at different hospitals may be duplicated or incomplete. Depending on the aggregation method used to combine results from the network, this can produce errors. For example, consider using a simple summation of aggregate counts: if a patient with hypertension receives medical care from both Hospital A and Hospital B, then it is possible that the sum will double count that patient, which results in the overestimation of the number of patients with hypertension (Weber, 2013).

Of course, this problem can be mostly alleviated by sending a hashed identifier of patients matching each hospital's queries to a trusted third party, but that again raises privacy concerns (Oechslin, 2003). There is some natural tradeoff between the privacy guaranteed to individual patients and the accuracy of the aggregate query, and hashed identifiers and simple summation fall at opposite ends of the spectrum. Several of the authors of this article recently proposed using the HyperLogLog (HLL) 'probabilistic sketch' (Durand and Flajolet, 2003; Flajolet and Martin, 1985; Flajolet *et al.*, 2007) to access intermediate tradeoffs of privacy versus accuracy (Yu and Weber, 2020). Probabilistic counting was introduced to the computer literature decades ago, and has found use in analyzing large streaming data in a variety of settings, ranging from internet routers (Cai *et al.*, 2005) to text corpora comparisons (Broder, 1997) to genomic sequences (Baker and Langmead, 2019; Ondov *et al.*, 2016; Solomon and Kingsford, 2018). Instead of sharing a single aggregate count, or sharing the full list of matching patient IDs (Weber, 2013), each hospital instead shares a smaller 'summary sketch' built from taking the logarithm of a coordinated random sample of m matching patient hashed IDs (Yu and Weber, 2020). Because only m patient IDs are used, and those are obfuscated through taking a logarithm, these HLL sketches are significantly more private than sending a full list of matching IDs. Due to the way the estimators work, HLL sketches have an error of about $1.04/\sqrt{m}$, which can be much less than expected from simple summation.

But when any data are shared by a hospital to a third party, there is risk of accidental leakage. Advances in homomorphic encryption and secure multi-party computation (Lindell, 2005) may eventually solve this problem by not allowing the third party any unencrypted data, but these are currently still impractical for deployment due to both computational and communication complexity. For example, consider the case where a hospital finds that there is only one patient satisfying the criterion for a query. If this hospital returns the aggregate count as one, then this unique patient's personal information is linked and can potentially be re-identified through a linkage attack (Emam and Dankar, 2008; Yu and Weber, 2020). To properly compare the privacy of various methods of data aggregation, we turn to the concept of k-anonymity. The basic idea behind k-anonymity is that if a method or dataset is k-anonymous, then each patient is similar to at least $k-1$ other patients with respect to potentially identifying variables, so that it is hard to determine the identity of a single patient in the dataset (Emam and Dankar, 2008; Sweeney, 2002). Although other mathematical formalisms like differential privacy (Dwork, 2008) are much stronger, they are harder to work with, as they require injecting deliberate noise, and are not currently widely in use by clinical databases. Furthermore, it is provably impossible for composable cardinality estimators (such as HLL) to be differentially private, because the ability to deduplicate runs counter to the base assumptions of differential privacy (Desfontaines *et al.*, 2019).

In this article, we will assume that hospitals in a federated network implement the HLL algorithm for queries. We will then prove bounds on the expected k-anonymity of the shared sketches, as well as provide fast algorithms for computing that expected k-anonymity. This study is an extension of previous work (Yu and Weber, 2020), which operated under the same setting and assumptions, but only provided empirical results and no proofs on the levels of privacy achieved. Here, we provide rigorous theoretical justification for those empirical claims.

2 Materials and methods

2.1 Setting and summary

In this article, we adopt the HLL sketch federated clinical network setting given in prior work (Yu and Weber, 2020). For completeness, we duplicate the salient points below.

Assume that every patient has a single invariant ID that is used across hospitals. Prototypically, one might consider using social security numbers in the USA for that purpose. Even without a single unique identifier, it is possible to generate an ID based off a combination of other possibly non-unique IDs, such as first and last name, zip code, address, birthdate, etc. Unfortunately, there may be errors in these records due to character recognition errors (e.g. S and 8), phonetic errors (e.g. ph and f) and typographic errors including insertion, transposition and substitutions. Luckily, there is a lot of existing literature on this problem, and methods such as BIN-DET and BIN-PROB (Durham *et al.*, 2010) have been proposed to deal with the issue. Thus, in this article, we will treat this problem as out-of-scope and assume for simplicity that every patient has a unique stable ID known to all institutions.

Further assume that there is a federated network of hospitals (or other institutions) responding to clinical queries, along with a central party that manages and relays messages. When hospitals receive a query, they generate a list of the IDs of patients who match the query. Each hospital will use a publicly known hash function to first pseudorandomly partition the patients into m buckets and then assign a uniform pseudorandom number between 0 and 1 to each patient. We also assume that the hash function is known by the attacker, because the attacker may have compromised one of the other hospitals or the central party. The hospital then stores the order of magnitude of the smallest number within each bucket, and sends these m smallest bucket values to the central party. By applying the HLL estimator, the central party is then able to compute the aggregate count for the query with a relative error of around $\frac{1.04}{\sqrt{m}}$ (Flajolet *et al.*, 2007).

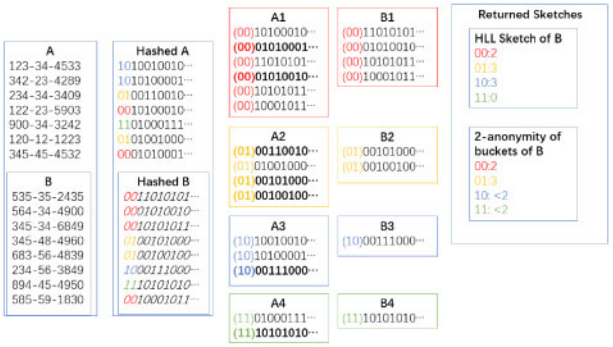

Fig. 1. Illustration of HyperLogLog k-anonymity. A hospital has an identified set B contained within the background population A. Binary hashes are taken of all patient identifiers. Those hashes are first used to partition the patients into four buckets. Within each bucket of B, the smallest value is chosen as the representative. Then the k-anonymity of that bucket is the number of hashes in the corresponding bucket of the background population that share the same position of the leading 1 bit

Here, we focus on an individual hospital and want to determine the expected exposure to accidentally disclosing private information if the central party is compromised. As the HLL sketch aggregates information within each of the m buckets, our goal is to compute the expected number of buckets which are not k-anonymized. In line with common practice, we set $k=10$ for most of our results, though the algorithms and proofs hold for other k. Below, we provide two approximation formulas for the expected value and in the Section 4 construct a table for the user to determine which approximation should be chosen based on the number of distinct patients and other relevant parameters.

2.2 k-Anonymity and HLL

2.2.1 High-level overview

The HLL (Flajolet *et al.*, 2007) probabilistic sketching algorithm is widely used to estimate the cardinality (number of different elements) of a set. Assume we have a database of electronic medical records; we can estimate the number of distinct patients by applying the HLL algorithm. The basic idea behind HLL is that the minimum value of a collection of random numbers between 0 and 1 is inversely proportional to the size of the collection. Therefore, we can estimate the cardinality of a set by first applying a hash function which maps all the elements uniformly onto $[0,1]$ and considering the minimum value. For the purposes of this article, we will operate in the random oracle model, where we assume that the hash function actually maps to a random number; in practice, a standard hash function like SHA-256 would probably be employed. In order to increase the accuracy of estimation, we randomly divide the set into m partitions and then estimate the cardinality of the original set by the harmonic mean from m partitions. Furthermore, the HLL algorithm only needs to store the position of the first 1 bit in the 64-bit hash value, rather than the full patient ID hash, providing partial privacy protection. As the expected error in the final estimate is around $1.04/\sqrt{m}$, increasing m can reduce the error of HLL query but increases the risk of privacy leaks.

In our setting, when a hospital is sent a query, there are two relevant sets to consider: (i) the background population (often, the set of all patients at the hospital) and (ii) the set of patients matching the query. The reason for considering the background population is that they can 'hide' patients who match the query by providing plausible deniability. The hospital will return a HLL sketch, which contains m values—the maximum position of the first 1 bit within each bucket. We define a HLL bucket with value x to be 'k-anonymous' if at least $k-1$ patients in the background population have hash value x; we will call these corresponding hash values in the background population hash *collisions* (Yu and Weber, 2020). This means that if an attacker gets access to the sketch and can narrow down the set of potential patients to the background population, they cannot determine with certainty which of the k patients with

that hash value was in the set of patients matching the query. Our goal is to determine the expected number of buckets that are not at least 10-anonymous (Fig. 1).

We wish to note that in this article, we deliberately use the much weaker notion of privacy provided by k-anonymity (Emam and Dankar, 2008), rather than stronger alternatives like differential privacy (Dwork, 2008), which have provable protection against inference attacks. Unfortunately, differential privacy (and similar alternatives) are provably incompatible with any composable cardinality estimation (Desfontaines *et al.*, 2019). In practice, hospital IRBs admit the use of 10-anonymity for query set patients as a useful metric, despite known issues with vulnerability of k-anonymity to inference attacks. Our article thus focuses on analyzing probabilistic sketches as a more private alternative to the standard practice of sending full hashed IDs.

2.2.2 Formal description

Let us recast the textual description above a bit more rigorously as the following mathematical problem:

Let A be a set and $B \subseteq A$ is a non-empty subset of A. A represents the background population and B represents patients satisfying the query. We define $r = \frac{|B|}{|A|}$ as the ratio of number of patients satisfying the query to background population (also sometimes known as concept prevalence).

Let $\sigma : A \to (0, 1]$ be a one-way hash function. In theory, we assume that we have a shared oracle available to both parties. In practice, a cryptographic hash function such as SHA-1, SHA-224 or SHA-256 (Johnson, 2020) is generally used. σ uniformly maps each element in A to a random real number in the interval $(0, 1]$.

Let $\bar{\sigma} : A \to \mathbf{Z}_{\geq 0}$ be defined by $\bar{\sigma}(x) = \lfloor -\log_2 \sigma(x) \rfloor$. This function returns the number of 0 bits before the first 1 bit in $x \in (0, 1]$ under a base 2 expansion.

Let $p : A \to \{1, .., m\}$ be a map that randomly partitions patients into m buckets. In practice, this map can also be derived from a cryptographic hash function. From the partition function p, we define $A_i = \{x \in A | p(x) = i\}$ and $B_i = B \cap A_i$, which, respectively, represent the ith bucket in whole database and sample.

Let $h_i(B) = \max\{\bar{\sigma}(x) | x \in B_i\}$ be the maximum number of zeros before the first one among all hash values represented in base 2 in the ith bucket of B which is B_i.

Let $e_i = \{x \in A_i | \bar{\sigma}(x) = h_i(B)\}$ be the set of elements in the ith bucket of A which collide with the elements in B_i.

We want to compute the $\mathbb{E}(|\{e_i | i = 1, \ldots, m$ and $0 < |e_i| \leq k - 1\}|)$, the expected number of non-k-anonymous buckets.

2.3 Probability of <k-anonymity without partition function

As described above, we need to consider the collisions against all m buckets. Here, however, we first show a simple analysis with no partition function (i.e. the case where $m = 1$) and compute the probability of each possible number of collisions so that in the later sections we can use this result to compute the desired expected value of 'non-k-anonymous' buckets.

Since there is only one bucket, there are only two sets A and B which represent the set of all patients and the set of patients matching the query, respectively. We denote $h(B) = \max\{\bar{\sigma}(x) | x \in B\}$, the maximum number of zeros before the first one among all hash values in base 2 in B, and $e = \{x \in A | \bar{\sigma}(x) = h(B)\}$, the set of collisions. We want to compute the probability that the number of collisions is less or equal to k, which is $P(|e| \leq k | A, B)$.

Each element in $\sigma(A)$ can be thought of as an i.i.d. random variable with distribution *Unif* $(0, 1)$. Therefore, $\bar{\sigma}(x) = n$ if and only if $\frac{1}{2^{n+1}} \leq \sigma(x) \leq \frac{1}{2^n}$. Then we get $P(\bar{\sigma}(x) = n) = \frac{1}{2^{n+1}}$. Thus, $P(h(B) = n) = \left(1 - \frac{1}{2^{n+1}}\right)^{|B|} - \left(1 - \frac{1}{2^n}\right)^{|B|}$.

Lemma 2.1. *Given sets $B \subset A$, the probability of exactly $n \in \mathbb{Z}_+$ collisions is:*

$$P(|e| = n | A, B) = \sum_{i=1}^{\infty} \sum_{k=1}^{\min(|B|, n)} f(i, k, |A|, |B|) g(i, n - k, |A|, |B|),$$

where $f(i, k, |A|, |B|) = \binom{|B|}{k} \left(\frac{1}{2^{i+1}}\right)^k \left(1 - \frac{1}{2^i}\right)^{|B|-k}$ and $g(i, k, |A|, |B|) = \binom{|A| - |B|}{k} \left(\frac{1}{2^{i+1}}\right)^k \left(1 - \frac{1}{2^{i+1}}\right)^{|A|-|B|-k}$.

Proof. Since the sets A and B are fixed, we use $P(|e| = n)$ to represent $P(|e| = n | A, B)$ for notational simplicity here.

By the law of total probability, we know that $P(|e| = n) = \sum_{i=1}^{\infty} \sum_{k=1}^{\min(|B|, n)} P(|e \cap B| = k \text{ and } h(B) = i) P(|e \cap (A - B)| = n - k \text{ and } h(B) = i)$.

First we consider the case where we have k collisions in $e \cap B$:

$$\begin{aligned} f(i, k, |A|, |B|) &= P(|e \cap B| = k \text{ and } h(B) = i) \\ &= \binom{|B|}{k} P(\bar{\sigma}(x_1) = \cdots = \bar{\sigma}(x_k) = i > \bar{\sigma}(x_{k+1}), \\ &\quad \cdots, \bar{\sigma}(x_{|B|})) \\ &= \binom{|B|}{k} \left(\frac{1}{2^{i+1}}\right)^k \left(1 - \frac{1}{2^i}\right)^{|B|-k}. \end{aligned}$$

Next we consider the case where we have k collisions in $e \cap (A - B)$:

$$\begin{aligned} g(i, k, |A|, |B|) &= P(|e \cap (A - B)| = k \text{ and } h(B) = i) \\ &= \binom{|A| - |B|}{k} P(\bar{\sigma}(x_1) = \cdots = \bar{\sigma}(x_k) = i \text{ and} \\ &\quad \bar{\sigma}(x_{k+1}), \ldots, \bar{\sigma}(x_{|A|-|B|}) \neq i) \\ &= \binom{|A| - |B|}{k} \left(\frac{1}{2^{i+1}}\right)^k (1 - \tfrac{1}{2^{i+1}})^{|A|-|B|-k} \end{aligned}$$

Thus,

$$P(|e| = n | A, B) = \sum_{i=1}^{\infty} \sum_{k=1}^{\min(|B|, n)} f(i, k, |A|, |B|) g(i, n - k, |A|, |B|) \quad (1)$$

and $P(0 < |e| \leq n) = \sum_{m=1}^{n} P(|e| = m)$ □

2.4 Expected number of buckets with less than k collisions

Recall that A is the background population and B the set of patients satisfying the query criteria. We denote the buckets of A and B under our partition function by $A_1, \ldots, A_m$ and $B_1, \ldots, B_m$ where $B_i = B \cap A_i$ for $i = 1, .., m$ and e_i is the sets of collisions in the ith bucket. Thus, the expected value of the number of buckets with no more than k collision is $E(|\{e_i | |e_i| \leq k, i = 1, \ldots, m\}|)$.

Note that $(|A_1|, \ldots, |A_m|) \sim \text{Multinomial}(|A|, p_1, \ldots, p_m)$ with $p_1 = \cdots = p_m = \frac{1}{m}$. Therefore, we know for a single bucket, say A_1, its cardinality follows a binomial distribution that is $|A_1| \sim \text{Binomial}\left(|A|, \frac{1}{m}\right)$

With a given A_i, $|B_i| \sim \text{Hypergeometric}(|A|, |A_i|, |B|)$. Thus,

$$P(|B_i| = b_i||A_i| = a_i) = \frac{\binom{a_i}{b_i}\binom{|A|-a_i}{|B|-b_i}}{\binom{|A|}{|B|}},$$

where $b_i \in \{0, 1, \ldots, \min(a_i, |B|)\}$ and $\mathbb{E}(|B_i|||A_i|) = \frac{|B|}{|A|}|A_i| = r|A_i| = \mu_b$ and $\text{Var}(|B_i|||A_i|) = r|A_i|\frac{|A|-|A_i|}{|A|}\frac{|A|-|B|}{|A|-1} = \sigma_b^2$.

Theorem 2.2.*The expected number of buckets which have at least 1 collision but no more than k collisions is:*

$$\mathbb{E}(k) = m\sum_{a=1}^{|A|} P_a \sum_{b=1}^{\min(a,|B|)} P_{k,a,b}P_{a,b},$$

where $P_a = P(|A_1| = a)$, $P_{a,b} = P(|B_1| = b||A_1| = a)$ *and* $P_{k,a,b} = P(0 < |e_1| \leq k||A_1| = a, |B_1| = b)$.

Proof.

$$\begin{aligned}
\mathbb{E}(k) &:= \mathbb{E}(|\{e_i|0 < |e_i| \leq k\}|)\\
&= \sum_{|A_1|,\ldots,|A_m|} [\mathbb{E}(|\{e_i|0 < |e_i| \leq k\}|||A_1|, \ldots, |A_m|)\\
&\qquad \times P(|A_1|, \ldots, |A_m|)]\\
&\qquad \text{(by the law of total expectation)}\\
&= \sum_{|A_1|,\ldots,|A_m|} [P(|A_1|, \ldots, |A_m|)\\
&\qquad \times \sum_{i=1}^{m} P(0 < |e_i| \leq k||A_1|, \ldots, |A_m|)]\\
&= m \sum_{|A_1|,\ldots,|A_m|} P(|A_1|, \ldots, |A_m|)P(0 < |e_i| \leq k||A_i|)\\
&\qquad \text{(by the independence of } |e_i| \text{ and } |A_j| \text{ for all } j \neq i)\\
&= m\sum_{|A_1|} P_{k,|A_1|} \sum_{|A_2|,\ldots,|A_m|} P(|A_1|, \ldots, |A_m|)\\
&\qquad \text{(by separating the summation)}\\
&= m\sum_{|A_1|} P_{k,|A_1|}P(|A_1|) \qquad \text{(by law of total probability)}
\end{aligned}$$

where $P_{k,|A_1|} = P(0 < |e_1| \leq k||A_1|)$.

In order to compute $P(0 < |e_1| \leq k||A_1|)$, we have to consider the range of $|B_1|$ which is $\{0, 1, \ldots, \min(|B|, |A_1|)\}$.

$$P(0 < |e_1| \leq k||A_1|) = \sum_{|B_1|} P(0 < |e_1| \leq k||A_1|, |B_1|)P(|B_1|||A_1|).$$

In contrast to the simple case in Section 2.3, here B_1 is not necessarily a proper subset of A_1 because A_1 can be the empty set and thus B_1 is also an empty set in this case. The collision number is zero if and only if A_1 is an empty sets. Therefore, we will expand the formula in Lemma 2.1 to compute $P(0 < |e_1| \leq k||A_1|, |B_1|)$. Furthermore, if we want rule out the case of zero collisions—because when the bucket is empty, there is not a patient ID for which we need to guarantee k-anonymity—we should set the range of $|A_1|$ and $|B_1|$ as $\{1, 2, \ldots, |A|\}$ and $\{1, 2, \ldots, \min(a, |B|)\}$, respectively.

$$P(|e_1| = k||A_1|, |B_1|) = \begin{cases} \text{equation (1)} & \text{if } k \neq 0\\ 1 & \text{if } k = 0 \text{ and } |B_1| = 0\\ 0 & \text{if } k = 0 \text{ and } |B_1| \neq 0. \end{cases}$$

Therefore, we will get:

$$\mathbb{E}(k) := m\sum_{a=1}^{|A|} P_a \sum_{b=1}^{\min(a,|B|)} P_{k,a,b}P_{a,b},$$

where $P_a = P(|A_1| = a) = \binom{A}{a}\left(\frac{1}{m}\right)^a\left(1-\frac{1}{m}\right)^{|A|-a}$, $P_{a,b} = P(|B_1| = b||A_1| = a) = \frac{\binom{a}{b}\binom{|A|-a}{|B|-b}}{\binom{|A|}{|B|}}$ and $P_{k,a,b} = P(|e_1| \leq k||A_1| = a, |B_1| = b)$. □

3 Algorithms

3.1 Time complexity of evaluating expectation

Again, recall that A is the background population, B is the set of patients satisfying the query criteria and e is the set of collisions. In Section 2.4, we gave an explicit formula for computing $P(|e| \leq k||A|, |B|)$. However, the time complexity of carrying out that computation is troublesome

$$\begin{aligned}
P(0 < |e| \leq k||A|, |B|) &= \sum_{n=1}^{k} P(|e| = n||A|, |B|)\\
&= \sum_{n=1}^{k}\sum_{i=1}^{\infty}\sum_{m=1}^{\min(|B|,n)} f(i,m)g(i,n-m),
\end{aligned}$$

where $f(i,m) = \binom{|B|}{m}\left(\frac{1}{2^{i+1}}\right)^m(1-\frac{1}{2^i})^{|B|-m}$ and $g(i,m) = \binom{|A|-|B|}{m}\left(\frac{1}{2^{i+1}}\right)^m(1-\frac{1}{2^{i+1}})^{|A|-|B|-m}$.

Usually, k is smaller than $|B|$ and the infinity in the second sum will be replaced by 64 (or some other constant <100) because it represents the maximum number of zeros before the first one among all hash values in base 2. As there are only 7 billion people on Earth, 64 bits is sufficient for the original hash function to have low probability of collisions. Therefore, the time complexity is $\mathcal{O}(k^2)$ for at most k collisions.

We consider the time complexity of computing the desired expectation. Theoretically, the range of $|A_1|$ is $\{1, 2, \ldots, |A|\}$ and the range of $|B_1|$ is $\{1, 2, \ldots, \min(|B|, |A_1|)\}$. Therefore, the computation time is:

$$\begin{aligned}
\sum_{|A_1|=1}^{|A|} \min(|A_1|, |B|) &= |B|(|A| - |B|) + \sum_{|A_1|=1}^{|B|} |A_1|\\
&= |B|(|A| - |B|) + \frac{|B|(|B|+1)}{2}\\
&= \left(r - \frac{1}{2}r^2\right)|A|^2 + \frac{r}{2}|A| \text{ where } r = \frac{|B|}{|A|},
\end{aligned}$$

and the time complexity is $\mathcal{O}(k^2|A|^2)$, which is quadratic in the size of population for at most k collisions. In practice, for large set sizes, it is computationally infeasible to use this theoretical formula to compute the desired expectation; thus, in the remainder of this article we analyze fast approximations.

3.2 Approximation A1: concentration inequalities

When $|A|$ is large, it is impossible to sum over whole range of $|A_1|$. Therefore, we will use concentration inequalities to restrict $|A_1|$ and $|B_1|$ to a smaller range. Because there is only an exponentially small probability that A_1 and B_1 will fall outside these restricted windows,

Table 1. Choice table for approximation method

			$r=0.1$		$r=0.08$		$r=0.05$		$r=0.01$		$r=0.005$		$r=0.001$	
$\lvert A\rvert/m$	$\lvert A\rvert$	m	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2
100	10^4	100	√		√		√		√		√		√	
50	10^4	200	√		√		√		√		√		√	
20	10^4	500	√		√		√		√		√		√	
1000	10^5	100	√		√		√		√		√		√	
500	10^5	200	√		√		√		√		√		√	
200	10^5	500	√		√		√		√		√		√	
100	10^5	1000	√		√		√		√		√		√	
50	10^5	2000	√		√		√		√		√		√	
20	10^5	5000	√		√		√		√		√		√	
10 000	10^6	100		√		√		√		√		√		√
2000	10^6	500		√		√		√	√		√		√	
1500	10^6	666		√		√		√		√	√		√	
1000	10^6	1000	√		√		√		√		√		√	
500	10^6	2000	√		√		√		√		√		√	
200	10^6	5000	√		√		√		√		√		√	
100	10^6	10 000	√		√		√		√		√		√	
50	10^6	20 000	√		√		√		√		√		√	
20	10^6	50 000	√		√		√		√		√		√	
100 000	10^7	100		√		√		√		√		√		√
20 000	10^7	500		√		√		√		√		√		√
10 000	10^7	1000		√		√		√		√		√		√
5000	10^7	2000		√		√		√		√		√		√
3333	10^7	3000		√		√		√		√		√		√
2000	10^7	5000		√		√		√		√	√		√	
1500	10^7	6666		√		√	√		√		√		√	
1000	10^7	10 000	√		√		√		√		√		√	
500	10^7	20 000	√		√		√		√		√		√	
200	10^7	50 000	√		√		√		√		√		√	

Note: A is the total size of the hospital background population, m is the number of buckets used in the HyperLogLog sketch and r is the fraction of the background population that matches the query criteria. 'A1' and 'A2', respectively, denote approximations 1 and 2. For every one of the parameter regimes, we used simulations to determine which of the approximation methods is more suitable for the practitioner.

this will have minimal effect on the final answer while reducing the computation time from quadratic to linear in the size of A.

Recall that $|A_1| \sim \text{Binomial}\left(|A|, \frac{1}{m}\right)$ and $\mathbb{E}(|A_1|) = \frac{|A|}{m} = \mu_a$, $\text{Var}(|A_1|) = \frac{|A|}{m} = \left(1 - \frac{1}{m}\right) = \sigma_a^2$. In order to reduce the time complexity, we will restrict $|A_1|$ in our computations to the interval $(L_a, U_a) := (\mu_a - 5\sigma_a, \mu_a + 5\sigma_a)$.

Recall that $|B_1| \sim \text{Hypergeometric}(|A|, |A_1|, |B|)$ for a given $|A_1|$ and $\mathbb{E}(|B_1||A_1|) = r|A_1| = \mu_b$, $Var(|B_1||A_1|) = r|A_1|\frac{|A|-|A_1|}{|A|}\frac{|A|-|B|}{|A|-1} = \sigma_b^2$. However, we define $\sigma_b'^2 = r|A_1|\frac{|A|-|B|}{|A|}$ which is greater than σ_b^2 and restrict $|B_1|$ in the interval $(L_b, U_b) = (\mu_b - 5\sigma_b', \mu_b + 5\sigma_b')$ in order to compute the error bound more easily below in Section 3.2.1. After concentration, we can make sure that $P(||A_1| - \mu_a| \geq 5\sigma_a) \lesssim 9.6 \times 10^{-4}$ and $P(||B_1| - \mu_b| \geq 5\sigma_b') \lesssim 9.6 \times 10^{-4}$ which is shown below in detail in Section 3.2.1. As an aside, while these two intervals of $|A_1|$ and $|B_1|$ have been chosen for analyzing the error bound and time complexity analytically, in the computing code we can directly use built-in functions to compute the relevant confidence intervals for $|A_1|$ and $|B_1|$.

By the concentration inequalities on $|A_1|$ and $|B_1|$, the desired expectation will be approximated by:

$$\mathbb{E}_1(k) := m \sum_{|A_1|=L_a}^{U_a} P_{|A_1|} \sum_{|B_1|=L_b}^{\min(U_b,|A_1|)} P_{k,|A_1|,|B_1|} P_{|A_1|,|B_1|},$$

where $P_{|A_1|,|B_1|} = P(|B_1||A_1|), P_{|A_1|} = P(|A_1|)$ and $P_{k,|A_1|,|B_1|} = P(0 < |e_1| \leq k||A_1|, |B_1|)$.

The computation time after concentration is:

$$\begin{aligned}
&\sum_{|A_1|=L_a}^{U_a} (U_b - L_b) \\
&= \sum_{|A_1|=L_a}^{U_a} 10\sigma_b' \\
&= \sum_{|A_1|=L_a}^{U_a} 10\sqrt{r(1-r)|A_1|} \\
&\leq 100\sqrt{r(1-r)}\sigma_a\sqrt{U_a} \\
&= 100\sqrt{r(1-r)}\sqrt{\frac{|A|}{m}\left(1-\frac{1}{m}\right)}\sqrt{\frac{|A|}{m} + 5\sqrt{\frac{|A|}{m}\left(1-\frac{1}{m}\right)}}.
\end{aligned}$$

So, the time complexity after concentration is $\mathcal{O}\left(k^2\frac{|A|}{m}\right)$ which is linear in $\frac{|A|}{m}$. After concentration, the expected value $E_1(k)$ is smaller than the actual $\mathbb{E}(k)$, but we can bound the error.

3.2.1 Error bounds

Recall that $|A_1| \sim \text{Binomial}(|A|, \frac{1}{m})$ and $\mu_a := \mathbb{E}(|A_1|) = \frac{|A|}{m}$, $\sigma_a^2 := \sqrt{|A|\frac{1}{m}\left(1-\frac{1}{m}\right)}$. We concentrate $|A_1|$ in the interval $(L_a, U_a) := (\mu_a - 5\sigma_a, \mu_a + 5\sigma_a)$. We define $F_a(x) := P(|A_1| \leq x)$ the cumulative density function of $|A_1|$.

First, we consider the concentration on $|A_1|$. We will apply the higher moments inequality on $|A_1| - \mu_a$ (Blum *et al.*, 2020):

$$P(||A_1| - \mu_a| \geq a) \leq \frac{\mathbb{E}(||A_1| - \mu_a|^r)}{a^r} \text{ for any positive even integer } r.$$

If we choose $r = 6$ then, we will get:

$$\begin{aligned} &P(||A_1| - \mu_a| \geq 5\sigma_a) \\ &\leq \frac{\mathbb{E}(||A_1| - \mu_a|^r)}{(5\sigma_a)^6} \\ &= \frac{1}{5^6}\left(\frac{1 - 30\delta + 120\delta^2 + 25|A|\delta - 130|A|\delta^2 + 15|A|^2\delta^2}{|A|^2\delta^2}\right) \\ &\leq \frac{15}{5^6} = 9.6 \times 10^{-4}, \end{aligned}$$

where $\delta = \frac{m-1}{m^2}$.

Then we consider the $|B_1|$ for a given $|A_1|$. For a given $|A_1|$, we know $|B_1| \sim$ Hypergeometric $(|A|, |A_1|, |B|)$ and $\mu_b := \mathbb{E}(|B_1|||A_1|) = r|A_1|, \sigma_b^2 := \mathrm{Var}(|B_1|||A_1|) = r|A_1|\frac{|A|-|A_1|}{|A|}\frac{|A|-|B|}{|A|-1}$. We concentrate $|B_1|$ in the interval $(L_b, U_b) := (\mu_b - 5\sigma_b', \mu_b + 5\sigma_b')$ where $\sigma_b'^2 = r|A_1|\frac{|A|-|B|}{|A|}$. We define $F_b(x) := P(|B_1| \leq x||A_1|)$ the cumulative density function of $|B_1|$ for a given $|A_1|$.

Note that for $X \sim \text{Binomial}(|A_1|, r)$, we can get $\mathbb{E}(X) = r|A_1|$ and $\mathrm{Var}(X) = r(1-r)|A_1|$. The expected value is equal to $\mathbb{E}(|B_1|||A_1|)$ and the variance is equal to $\sigma_b'^2$ which is bigger than the variance of $|B_1|$ for this given $|A_1|$. This explains that the hypergeometric distribution is more concentrated about the mean than the binomial distribution (Kalbfleisch, 1985). Therefore, we will use this binomial distribution to bound the tail of our hypergeometric distribution:

$$\begin{aligned} P(L_b \geq |B_1| or |B_1| \geq U_b) &\leq P(L_b \geq X \text{ or } X \leq U_b) \\ &= P\left(|X - \mathbb{E}(X)| \geq 5\sqrt{\mathrm{Var}(X)}\right) \\ &\lesssim 9.6 \times 10^{-4}, \end{aligned}$$

$$\begin{aligned} &\mathbb{E}(k) - \mathbb{E}_1(k) \\ &= m[\sum_{\substack{|A_1| < L_a \\ |A_1| > U_a}} P_{|A_1|} \sum_{|B_1|} P_{k,|A_1|,|B_1|} P_{|A_1|,|B_1|} \\ &+ \sum_{|A_1|=L_a}^{U_a} P_{|A_1|} \sum_{\substack{|B_1| < L_b \\ |B_1| > U_b}} P_{k,|A_1|,|B_1|} P_{|A_1|,|B_1|}] \\ &\leq m[\sum_{\substack{|A_1| < L_a \\ |A_1| > U_a}} P_{|A_1|} + \sum_{|A_1|=L_a}^{U_a} P(|A_1|) \sum_{\substack{|B_1| < L_b \\ |B_1| > U_b}} P_{|A_1|,|B_1|}] \\ &\leq m1.92 \times 10^{-3}, \end{aligned}$$

where $P_{|A_1|,|B_1|} = P(|B_1|||A_1|), P_{|A_1|} = P(|A_1|)$ and $P_{k,|A_1|,|B_1|} = P(|e_1| \leq k||A_1|, |B_1|)$.

But the in the computing code, we can use the built-in function to find the interval (L_a, U_a) and (L_b, U_b) such that $P(L_a \leq |A_1| \leq U_a) \geq 1 - \alpha$ and $P(L_b \leq |B_1| \leq U_b) \geq 1 - \alpha$. This will not affect the time complexity and can ensure that the absolute error between the estimated expected value and the actual expected value is <1 by choosing a proper α. It is obvious the smaller α is, the smaller the error will be, but the intervals (L_a, U_a) and (L_b, U_b) will be bigger which means a longer computing time. Therefore, there is a tradeoff between accuracy and speed (see Table 2 for real computing time). Fortunately, in all cases we explore, the L_a and U_a given above can ensure that $\alpha < 5 \times 10^{-5}$.

$$\begin{aligned} &\mathbb{E}(k) - \mathbb{E}_1(k) \\ &= m[\sum_{\substack{|A_1| < L_a \\ |A_1| > U_a}} P_{|A_1|} P(|e_1| \leq k||A_1|) \\ &+ \sum_{|A_1|=L_a}^{U_a} P_{|A_1|} \sum_{\substack{|B_1| < L_b \\ |B_1| > U_b}} P_{k,|A_1|,|B_1|} P_{|A_1|,|B_1|}] \\ &\leq m[\sum_{\substack{|A_1| < L_a \\ |A_1| > U_a}} P_{|A_1|} + \sum_{|A_1|=L_a}^{U_a} P_{|A_1|} \sum_{\substack{|B_1| < L_b \\ |B_1| > U_b}} P_{|A_1|,|B_1|}] \\ &\leq 2m\alpha, \end{aligned}$$

where $P_{|A_1|,|B_1|} = P(|B_1|||A_1|), P_{|A_1|} = P(|A_1|)$ and $P_{k,|A_1|,|B_1|} = P(0 < |e_1| \leq k||A_1|, |B_1|)$.

3.2.2 Approximation A2: mean-field approximation

Although the time complexity after concentration is linear in $\frac{|A|}{m}$, for large $|A|$ and m small, this speedup is often still not enough. We can further approximate $P(|e_1| \leq k||A_1|, |B_1|)$ by $P(|e_1| \leq k||A_1|, |B_1| = r|A_1|)$ and get the following approximation of the expectation:

$$\begin{aligned} \mathbb{E}_2(k) := m \sum_{|A_1|=L_a}^{U_a} P(|A_1|) \sum_{|B_1|=L_b}^{U_b} [P(|e_1| \leq k||A_1|, r|A_1|) \\ \times P(|B_1|||A_1|)] \\ = m \sum_{|A_1|=L_a}^{U_a} P(|A_1|) P(|e_1| \leq k||A_1|, r|A_1|). \end{aligned}$$

This is a 'mean-field' approximation based on Approximation A1. The basic idea behind this approximation is to use the probability at the mean value which is $P(0 < |e_1| < k||A_1|, r|A_1|)$ to represent all the probabilities $P(0 < |e_1| < k||A_1|, |B_1|)$ when $|B_1| \in (L_b, U_b)$ because $P(0 < |e_1| < k||A_1|, |B_1|)$ is monotonic increasing in $|B_1|$ and the interval (L_b, U_b) is small enough compared with the theoretical range $(0, \min(|A_1|, |B|))$.

The range of $|A_1|$ is still $(L_a, U_a) = (\mu_a - 5\sigma_a, \mu_a + 5\sigma_a)$. Therefore, the computation time of E_2 is:

$$(U_a - L_a) = 10\sigma_a = 10\sqrt{\frac{|A|}{m}\left(1 - \frac{1}{m}\right)},$$

and the time complexity is $\mathcal{O}\left(k^2\sqrt{\frac{|A|}{m}}\right)$. The real computing time will be discussed in the Section 4. Unfortunately, we do not have a strong provable guarantee with this approximation, but it seems empirically to work well in practice.

4 Results

In order to assess the accuracy–speed tradeoffs of our two approximations, we ran simulations measuring the ground truth empirical k-anonymity of patients in several different regimes using HLL sketches. Those simulations serve as the ground truth since they have the same distribution as hashing real patient identifiers with a random seed, without needing to use real patient data for this article. Then, we compared those empirical values against the approximations described in this article. In the large cardinality regimes, it is computationally infeasible to run full simulations, so we only

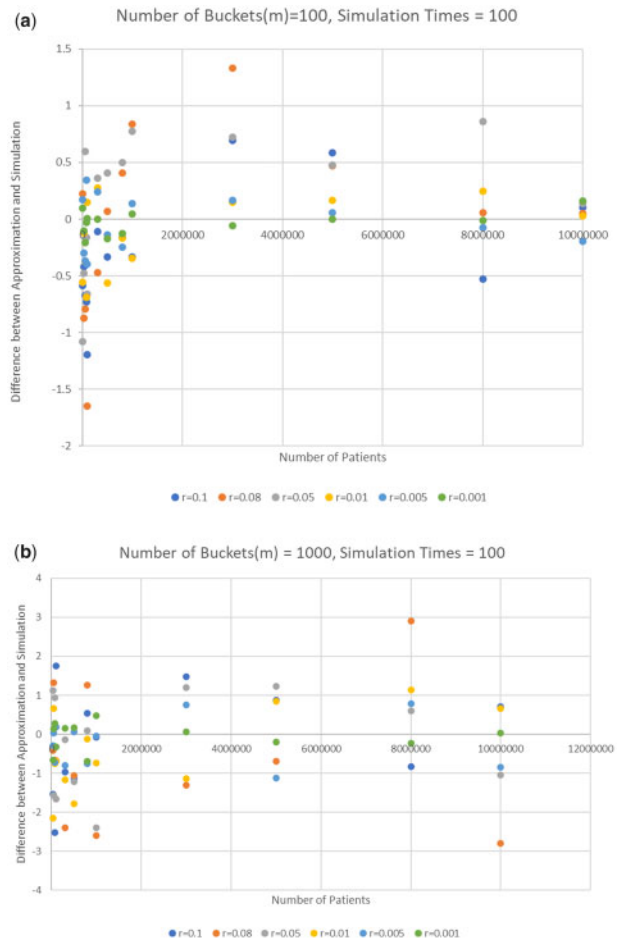

Fig. 2. Errors between Approximation (based on choice table) and simulation of 100 random trials with number of buckets = 100 (top) and 1000 (bottom)

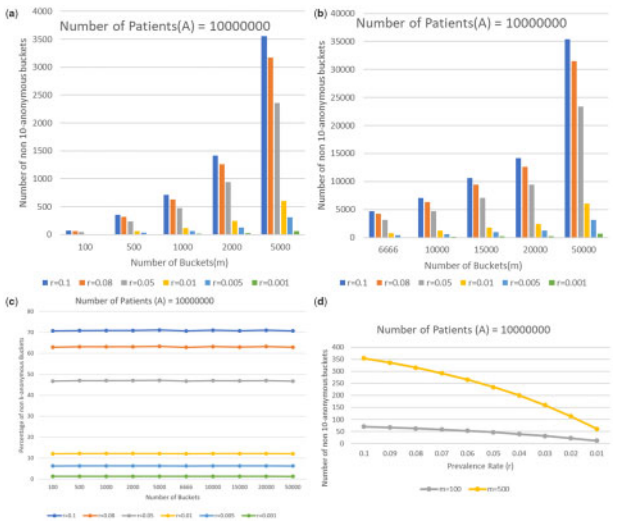

Fig. 3. Expected number of non-10-anonymous buckets under different combinations of number of buckets (m) and prevalence rate (r) when total number of patients is 10^7. (Top) Number of non-10-anonymous buckets under different combinations of m (number of buckets) and r (prevalence rate) when total number of patients is 10^7. (Left bottom) However, the fraction of non-10-anonymous buckets remains constant as the number of buckets increase when the other variables are held fixed. (Right bottom) It is the relationship to prevalence rate that is more complicated and nonlinear, as shown by focusing on the behavior for 100 and 500 buckets

compare the run-times of the two approximation methods. In Table 2, we provide full tables of these results. In Table 1, we provide a high-level summary giving a practitioner guidance on which method is appropriate under those particular parameter choices. All computations were run in single-thread mode on an AMD Ryzen Threadripper 3970X 32-core CPU machine running Ubuntu 18.04.5 LTS (bionic) with 256 GiB of RAM. It is worth mentioning that steps in computations are trivially parallelizable, but for benchmarking purposes all our results are of single-threaded performance. Additionally, instead of using actual hash functions (e.g. SHA-256), we generate uniform random numbers as the hashed values, which has the same probability distribution. Code is available on Github and relies on using the numpy, scipy.stat and decimal packages for simulation of patient hashes and explicit computation of probability distributions: https://github.com/tzyRachel/K-anonymity-Expectation

Recall that A represents the number of all patients, B represents the number of patients who meet some query criteria and m is the number of buckets in the HLL process. We introduce $r = \frac{|B|}{|A|}$ to represent the ratio of $|A|$ and $|B|$, because as we will see, this ratio controls to a large extent the number of collisions. Intuitively, r represents the number of background population persons who could be used to provide plausible deniability to each patient in the query set.

Our simulations sweep over the different combinations of the parameters A, r and m to construct a table to fit Approximations A1 and A2. In all simulations, we restrict $|A|$ in the interval $[10^4, 10^7]$ and m in the interval $[100, 50\,000]$. In this article, the total number of different patients in a hospital is assumed to be over 10^4; not only are approximation methods unnecessary for $|A| < 10^4$ because exact computations are feasible, but there is not a sufficiently large background population to hide the query set when $|A|$ is small, and the privacy characteristics then become equivalent to sending hashed IDs (Yu and Weber, 2020). Since the simulations are run under the condition of '10-anonymity', we make sure that $\frac{|A|}{m} > 20$ which is the mean value of the single bucket size. Also, r is restricted in the interval $[0.001, 0.1]$ and we choose six different values of r which are $0.1, 0.08, 0.05, 0.01, 0.005, 0.001$ to run the simulations and compare the simulation results with computing results.

As we discussed in the Section 2, we can estimate the desired expected value by both Approximations A1 and A2. The final choice of Approximation A1 or Approximation A2 seems to be dependent primarily on $\frac{|A|}{m}$. In most cases, when $\frac{|A|}{m} \geq 1500$, Approximation A2 is good enough and the computing time is no longer than 3 min. When $\frac{|A|}{m} < 1500$, Approximation A2 will be not accurate enough and we have to choose Approximation A1. The computing time of Approximation A1 is proportional to $\sqrt{r\frac{|A|}{m}}$, which is sometimes a concern. When $\frac{|A|}{m} \leq 1500$, the computing time is usually no longer than 8 min. But there are several special cases, such as when $r = 0.1$ and $r = 0.08$, that the computing time at $A = 10^7, m = 6666$ is $\sim$10 min which might be acceptable but is really not ideal. Furthermore, in extreme cases, the approximate expected k-anonymity return by Approximations 1 and 2 differ by $\sim$10 (Table 2).

To make things easier for the end-practitioner, we provide a summary 'choice' table (Table 1) guiding them on which approximation is suggested, based on different numbers of patients, numbers of buckets and ratios of number of patients matching query to all patients. Choosing between approximations A1 and A2 is an accuracy/running time tradeoff. A1 is usually both more accurate and expensive than A2. For the purpose of the choice table, to give a concrete recommendation, we aim to have single-threaded running

Table 2 Expected number of non-10-anonymous buckets from Approximations A1 and A2 compared against ground truth simulations

$\|A\|/m$	$\|A\|$	m	r	Simulation average	Simulation replicates	A1	A1 time (s)	A2	A2 time (s)
100	10 000	100	0.1	70.60	100	70.28	14.75	72.76	2.00
50	10 000	200	0.1	141.14	100	141.12	6.61	149.85	1.00
20	10 000	500	0.1	354.38	100	353.74	2.60	414.61	0.20
300	30 000	100	0.1	70.68	100	70.60	62.26	71.60	3.00
150	30 000	200	0.1	141.79	100	141.59	26.66	144.55	2.00
60	30 000	500	0.1	354.90	100	354.65	9.88	372.67	1.00
30	30 000	1000	0.1	712.22	100	709.73	4.12	783.81	0.40
500	50 000	100	0.1	71.87	100	70.71	84.74	71.44	4.00
250	50 000	200	0.1	142.94	100	141.76	47.65	143.68	3.00
100	50 000	500	0.1	352.96	100	353.20	19.70	363.80	2.00
50	50 000	1000	0.1	707.75	100	706.99	8.19	749.25	0.70
800	80 000	100	0.1	70.57	100	70.40	136.09	70.98	4.00
400	80 000	200	0.1	142.29	100	140.90	77.41	142.45	3.00
160	80 000	500	0.1	354.54	100	353.85	34.99	360.40	2.00
80	80 000	1000	0.1	704.88	100	707.91	16.47	734.01	1.00
1000	100 000	100	0.1	71.13	100	70.69	252.00	71.24	4.60
500	100 000	200	0.1	142.77	100	141.76	134.00	142.87	3.18
200	100 000	500	0.1	354.27	100	353.37	40.57	358.63	2.00
100	100 000	1000	0.1	705.02	100	706.95	22.16	727.60	1.30
50	100 000	2000	0.1	1416.61	100	1414.37	9.95	1498.50	0.70
20	100 000	5000	0.1	3536.27	100	3539.54	3.37	4146.33	0.20
3000	300 000	100	0.1	70.47	100			70.76	8.00
300	300 000	1000	0.1	709.57	100	709.02	90.00	715.96	2.40
5000	500 000	100	0.1	71.13	100			70.91	10.00
500	500 000	1000	0.1	708.77	100	710.08	155.00	714.36	3.00
8000	800 000	100	0.1	71.92	100			71.06	14.00
800	800 000	1000	0.1	708.00	100	707.00	25.00	709.76	4.00
10 000	1 000 000	100	0.1	70.58	100			70.89	16.00
2000	1 000 000	500	0.1	356.33	100	354.85	607.00	355.69	7.00
1000	1 000 000	1000	0.1	707.66	100	710.06	316.00	712.36	5.00
500	1 000 000	2000	0.1	1419.32	60	1420.47	150.00	1428.72	3.00
200	1 000 000	5000	0.1	3534.64	50	3536.36	65.00	3586.25	2.00
100	1 000 000	10 000	0.1	7068.96	50	7073.09	30.00	7275.97	1.30
50	1 000 000	20 000	0.1			14 146.62	12.00	14 985.00	0.70
20	1 000 000	50 000	0.1			35 396.39	4.00	41 463.50	0.20
30 000	3 000 000	100	0.1	71.01	100			70.98	30.00
3000	3 000 000	1000	0.1	703.79	100			707.55	8.00
50 000	5 000 000	100	0.1	71.54	100			70.71	40.00
5000	5 000 000	1000	0.1	708.32	100			709.07	12.00
80 000	8 000 000	100	0.1	71.01	100			70.87	50.00
8000	8000 000	1000	0.1	707.81	70			710.63	15.00
1 00 000	10 000 000	100	0.1	70.48	100			70.71	55.00
20 000	10 000 000	500	0.1	354.08	100			354.39	30.00
10 000	10 000 000	1000	0.1	711.81	70			708.87	16.00
5000	10 000 000	2000	0.1					1418.13	11.00
2000	10 000 000	5000	0.1			3551.59	726.00	3556.85	7.00
1500.15	10 000 000	6666	0.1			4711.94	547.00	4720.90	5.70
1000	10 000 000	10 000	0.1			7103.56	366.86	7123.63	4.60
666.7	10 000 000	15 000	0.1			10 614.93	250.00	10 659.18	3.70
500	10 000 000	20 000	0.1			14 207.49	192.00	14 287.16	3.00
200	10 000 000	50 000	0.1			35 366.18	79.00	35 862.54	2.00

Note: Some entries are empty because the computation time was infeasibly long. We have highlighted (in yellow or green) the more accurate approximation finished within 10 min. Full simulation and computation results for $r \in \{0.1, 0.08, 0.05, 0.01, 0.005, 0.001\}$ are available on Github in machine-readable format.

times below 10 min; many modern multi-core machines can run over a dozen threads at once, and given that the approximation algorithms are trivially parallelizable, this amounts implicitly to a goal of real wall-clock time of less than a minute. The choice table is filled out by selecting the approximation method with the least error given that time constraint. In most cases, we choose A1 if the running time is below 10 min. Sometimes, computation results from A1 and A2 are almost the same, so the faster method can be chosen. Based on this rule, we compare gold-standard simulation results against the approximations in Table 2 to construct the 'choice' table. Note that our choice of 10 min single-threaded run-time was arbitrary; given extra computational resources, the ideal switch-off point between approximations will vary.

Figure 2 shows the errors between the approximation results (based on the choice Table 1) and simulation results (Table 2) when number of distinct patients is 10^7 and number of buckets are 100 and 1000, respectively. The absolute values of all the errors are no more than 4.

5 Discussion

We first note that all of the approximations we have provided finish on the order of minutes. As they are analytical approximations, there is also no need to run them multiple times. Although we have not shown explicit simulation run-times in the tables above, the larger simulations take upwards of hours; furthermore, we did not perform simulations for the largest parameter ranges because we expected those to take significantly longer. Our approximations speed up determining the expected privacy loss from distributing HLL sketches.

We are also able to form some general conclusions about the expected privacy of HLL sketches. As mentioned above the prevalence ratio $r = \frac{|B|}{|A|}$, where A and B are, respectively, the background population and query population can be interpreted as the ratio of patients matching a query (e.g. 'How many patients have been diagnosed with diabetes?'). Based on HLL, m is the number of buckets and A_i and B_i are the ith bucket in A and B. Figure 3 plots the number of buckets and prevalence rate against the estimated expected number of non-'k-anonymized' buckets and the number of buckets versus the percentage of the non-'k-anonymous' buckets. The two top plots are simply the number of non-k-anonymous buckets against the number of buckets and varying the other parameters, but this turns out to not be the right set of variables to control.

Instead, as evidenced by the lower-left plot (Fig. 3), a roughly constant fraction of the buckets are not k-anonymized when r is constant. This is unsurprising because as mentioned earlier, r is intuitively the number of background population members that could be used to hide each patient. Of course, random chance also plays a large role. More precisely, this constant is close to $100P(0 < |e| < 10|\frac{|A|}{m}, r\frac{|A|}{m})$ where $P(0 < |e| < 10|\frac{|A|}{m}, r\frac{|A|}{m})$ is the probability of that the number of collisions is >0 and <10 when the bucket size is at the mean value $\frac{|A|}{m}$. It is not quite equal for two reasons. The first reason is the obvious one, that we are using the approximations that form the subject of this article. The second reason is that the single bucket size $|A_1|$ follows a Binomial distribution with mean $\frac{|A|}{m}$ and $p = \frac{1}{m}$. When $|A|$ and $\frac{|A|}{m}$ are big enough, we can get $P(0 < |e_1| \leq 10||A_1|, |B_1|) \approx P(0 < |e_1| \leq 10|\frac{|A|}{m}, r\frac{|A|}{m})$ by concentrating $|A_1|, |B_1|$ in an interval centered at the means, which is similar to what we did in Approximation A2, but simpler. However, when $\frac{|A|}{m}$ is not that big, for example, $|A| = 100, m = 5$, then $P(0 < |e_1| \leq 10|\frac{|A|}{m}, r\frac{|A|}{m})$ and $P(0 < |e_1| \leq 10||A_1|, r|B_1|)$ will differ a lot for different value of $|A_1|$ and $|B_1|$.

Now that it is clear that r is the value of primary importance, we see in the lower-right plot of Figure 3 that as prevalence rate (r) increases, more buckets are non-'k-anonymized'. This is because bigger r means more overlap between sets A and B and also each pair of buckets A_i and B_i. Thus, the maximum number of zeros before the first one among all hash values in B_i is more likely equal to that in A_i. Thus, a hospital IRB or clinical query system seeking to understand the 10-anonymity of a particular query can use a first-order approximation based only on r, without even needing to run our code. Indeed, they need only consult our lower-right plot in Figure 3 and scale to the size of their background population to determine that first-order approximation. This can be done without any code. When a more precise result is needed, however, our two Approximations can provide that answer in only a few minutes. Of course, if even that is insufficient, the practitioner may choose to directly measure the k-anonymity of a particular HLL sketch; this is not in the scope of this article, but was empirically done in prior work (Yu and Weber, 2020).

6 Conclusion

In this article, we have developed a method to quickly compute the expected number of non-'k-anonymous' buckets in the HLL sketch. Because of the number of patients (denoted as $|A|$ in our model) is too big to compute the precise expected value, we introduced two approximations based on concentration inequalities. In general, Approximation A1 is suitable for the case when the expected value of single bucket size which is $\frac{|A|}{m}$ is 'small', for example, total number of patients ($|A|$) is 10^5 and number of buckets (m) is 100 or total number of patients ($|A|$) is 10^7 and number of buckets (m) is 10^5. Approximation A2 is suitable for the case when $\frac{|A|}{m}$ is 'big', for example, total number of patients ($|A|$) is 10^7 and number of buckets (m) is 100 (see choice table in Section 4).

By an appropriate choice of approximation method, we can control the computing time to under 300 s in almost all the cases. In other words, when an individual hospital is asked a query to return the aggregate counts based on sharing HLL sketches, we can compute the expected number of buckets which match fewer than 10 patients in the background population. If this number is too high, that is a signal to the clinical query system that the particular query is unsafe to release using HLL sketches. It is then up to the clinical query system to decide whether to fall back on another aggregation method, or if they should simply not respond to the query.

Our results further give some guidance into the parameter ranges in which HLL sketches are likely to be safe to release. HLL sketches are especially useful for rare diseases, where the prevalence ratio in the population is low. Note that this is in marked contrast to sending raw counts, where rare diseases are precisely the least k-anonymous. Thus, HLL sketches fill a complementary role. Indeed, at the heart of the problem is the tradeoff between the utility/accuracy of HLL sketches and privacy, which increase or decrease, respectively, with the number of buckets. The average k-anonymity of a bucket is roughly inversely proportional to the square of the estimation error; our work computes instead the number of buckets that are not at least 10-anonymous. For more guidance on this tradeoff, we refer the reader to prior work, where we graphed this tradeoff empirically (Yu and Weber, 2020).

Ultimately, our work is primarily useful in contexts where federated clinical query systems are used in biomedical research. The past year has seen increasing amounts of data centralization to combat the Covid-19 pandemic. The cost to privacy has been accepted because of the urgent clear and present need. However, in the future post-pandemic era as the pendulum swings the other direction, privacy may again take center stage. We hope that our work will be useful in analyzing the privacy consequences of distributed query systems and help inform policy-makers and institutional IRBs about the privacy-utility tradeoffs at hand.

Data Availability

All of the data and code used to generate benchmarks is available on the Github: https://github.com/tzyRachel/K-anonymity-Expectation

Funding

We acknowledge startup funding from the University of Toronto Department of Computer and Mathematical Sciences for support.

Conflict of Interest: none declared.

References

Baker,D.N. and Langmead,B. (2019) Dashing: fast and accurate genomic distances with HyperLogLog. *Genome Biol.*, **20**, 265.

Bengio,Y. *et al.* (2020) The need for privacy with public digital contact tracing during the COVID-19 pandemic. *Lancet Digit. Health*, **2**, e342–e344.

Bengio,Y. *et al.* (2021) Inherent privacy limitations of decentralized contact tracing apps. *J. Am. Med. Inform. Assoc.*, **28**, 193–195.

Benitez,K. and Malin,B. (2010) Evaluating re-identification risks with respect to the HIPAA privacy rule. *J. Am. Med. Inform. Assoc.*, **17**, 169–177.

Blum,A. *et al.* (2020) *Foundations of Data Science*. Cambridge University Press, Cambridge.

Brat,G.A. *et al.* (2020) International electronic health record-derived COVID-19 clinical course profiles: the 4CE consortium. *NPJ Digit. Med.*, **3**, 1–9.

Broder,A.Z. (1997) On the resemblance and containment of documents. In: *Proceedings Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, IEEE, pp. 21–29.

Cai,M. *et al.* (2005) Fast and accurate traffic matrix measurement using adaptive cardinality counting. In: *Proceeding of the 2005 ACM SIGCOMM Workshop on Mining Network Data - MineNet '05*, pp. 205–206.

Davies,M. *et al.* (2016) Software-enabled distributed network governance: the PopMedNet experience. *eGEMs*, **4**, 5.

Desfontaines,D. *et al.* (2019) Cardinality estimators do not preserve privacy. *Proc. Priv. Enh. Technol.*, **2019**, 26–46.

Durand,M. and Flajolet,P. (2003) Loglog counting of large cardinalities. In: *Algorithms - ESA 2003 Lecture Notes in Computer Science*, pp. 605–617.

Durham,E. *et al.* (2010) Private medical record linkage with approximate matching. *AMIA Annu. Symp. Proc.*, **2010**, 182–186.

Dwork,C. (2008) Differential privacy: a survey of results. In: International Conference on Theory and Applications of Models of Computation, Springer, pp. 1–19.

Emam,K.E. and Dankar,F.K. (2008) Protecting privacy using k-anonymity. *J. Am. Med. Inform. Assoc.*, **15**, 627–637.

Emam,K.E. *et al.* (2009) Evaluating the risk of re-identification of patients from hospital prescription records. *Can. J. Hosp. Pharm.*, **62**, 307–319.

Flajolet,P. and Martin,G.N. (1985) Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.*, **31**, 182–209.

Flajolet,P. *et al.* (2007) HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm. In: Jacquet, P. (ed.) *Discrete Mathematics and Theoretical Computer Science 2007 Jun 17 (pp. 137-156). Discrete Mathematics and Theoretical Computer Science*, Juan les pins, France, pp. 127–146.

Fleurence,R.L. *et al.* (2014) Launching PCORnet, a national patient-centered clinical research network. *J. Am. Med. Inform. Assoc.*, **21**, 578–582.

Haendel,M.A. *et al.*; the N3C Consortium (2021) The National COVID Cohort Collaborative (n3c): rationale, design, infrastructure, and deployment. *J. Am. Med. Inform. Assoc.*, **28**, 427–443.

Heatherly,R.D. *et al.* (2013) Enabling genomic-phenomic association discovery without sacrificing anonymity. *PLoS One*, **8**, e53875.

Johnson,L. (2020) *Security Controls Evaluation, Testing, and Assessment Handbook*. Academic Press.

Kalbfleisch,J.G. (1985) *Probability and Statistical Inference*. Springer-Verlag.

Kho,A.N. and Goel,S. (2019) Systems and methods for enabling data de-identification and anonymous data linkage. *US Patent*, **10**, 454–901.

Lindell,Y. (2005) Secure multiparty computation for privacy preserving data mining. In Encyclopedia of Data Warehousing and Mining, IGI global, pp. 1005–1009.

Oechslin,P. (2003) Making a faster cryptanalytic time-memory trade-off. In: Boneh, D. (ed.) *Advances in Cryptology - CRYPTO 2003*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 617–630.

Ondov,B.D. *et al.* (2016) Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biol.*, **17**, 132.

Solomon,B. and Kingsford,C. (2018) Improved search of large transcriptomic sequencing databases using split sequence bloom trees. *J. Comput. Biol.*, **25**, 755–765.

Sweeney,L. (2002) k-Anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.*, **10**, 557–570.

Weber,G.M. (2013) Federated queries of clinical data repositories: the sum of the parts does not equal the whole. *J. Am. Med. Inform. Assoc.*, **20**, e155–e161.

Weber,G.M. (2015) Federated queries of clinical data repositories: scaling to a national network. *J. Biomed. Inform.*, **55**, 231–236.

Weber,G.M. *et al.* (2009) The Shared Health Research Information Network (SHRINE): a prototype federated query tool for clinical data repositories. *J. Am. Med. Inform. Assoc.*, **16**, 624–630.

Wyatt,M.C. *et al.* (2014) Federated Aggregate Cohort Estimator (FACE): an easy to deploy, vendor neutral, multi-institutional cohort query architecture. *J. Biomed. Inform.*, **52**, 65–71.

Yu,Y.W. and Weber,G.M. (2020) Balancing accuracy and privacy in federated queries of clinical data repositories: algorithm development and validation. *J. Med. Int. Res.*, **22**, e18735.