K-Randomization

Maxim Zhilyaev David Zeber

January 12, 2016

1 Differential Privacy

The typical setting for differential privacy is as follows. We consider a **database** as a collection of records. Each record is an element of some space $\mathcal{D}$, and a database $\mathbf{x}$ is a vector of n records: $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{D}^n$.

We wish to release information retrieved from the database by means of a **query**, a function A mapping the database into another space: $A : \mathcal{D}^n \to \mathcal{S}$. The result of applying a query to a database is termed a **transcript**. The query usually applies some aggregation to the database records, and so the output space $\mathcal{S}$ is generally of lower dimensionality than the original database. If the query is randomized, i.e., $A(\mathbf{x}) = A(\mathbf{x}; \xi)$ for a random element ξ, then the transcript will be a random element of $\mathcal{S}$.

The notion of differential privacy for a database query is that the resulting transcript does not change substantially when a record in the database is modified, i.e., transcripts are not sensitive to particular individual records in the database. Hence, releasing query transcripts publicly will not jeopardize privacy, since information regarding individual records cannot be gained by analyzing query transcripts.

Differential privacy for a randomized query A is formulated by comparing the transcripts generated by applying A to two very similar databases $\mathbf{x}, \mathbf{x}' \in \mathcal{D}^n$. We say the databases **differ in one row** if $\sum_{i=1}^{n} I(x_i \neq x_i') = 1$.

Definition. A randomized query A is ϵ**-differentially private** if, for any two databases $\mathbf{x}, \mathbf{x}' \in \mathcal{D}^n$ differing in one row,

$$\mathsf{P}[A(\mathbf{x}) \in S] \leq \exp(\epsilon) \cdot \mathsf{P}[A(\mathbf{x}') \in S] \tag{1.1}$$

for all $S \subset \mathcal{S}$ (measurable).

In other words, the transcripts from the two databases databases differing in one row are close in distribution. An alternative notion of differing in one row that is sometimes used is that $\mathbf{x}'$ includes an additional record that is not in $\mathbf{x}$: $\mathbf{x} \in \mathcal{D}^n$, $\mathbf{x}' \in \mathcal{D}^{n+1}$, and $x_i = x_i'$ for $i = 1, \ldots, n$.

If $\mathcal{S}$ is finite, which is common in cases where the transcript involves integer counts, then the distribution of the transcript $A(\mathbf{x})$ can be represented using its pmf $\mathsf{P}[A(\mathbf{x}) = s]$ for $s \in \mathcal{S}$. In this case, the differential privacy condition can also be expressed in terms of the pmf.

Proposition 1.1. *If $\mathcal{S}$ is finite, then A is ϵ-**differentially private** if and only if*

$$\mathsf{P}[A(\mathbf{x}) = s] \leq \exp(\epsilon) \cdot \mathsf{P}[A(\mathbf{x}') = s] \tag{1.2}$$

for all $s \in \mathcal{S}$, where $\mathbf{x}, \mathbf{x}'$ differ in one row.

Proof. ($\Leftarrow$) Given $S \subset \mathcal{S}$, we can write $\mathsf{P}[A(\mathbf{x}) \in S] = \sum_{s \in S} \mathsf{P}[A(\mathbf{x}) = s]$. If $\mathsf{P}[A(\mathbf{x}') \in S] = 0$, then $\mathsf{P}[A(\mathbf{x}') = s] = 0$ for each $s \in S$. From (1.2) we have that $\mathsf{P}[A(\mathbf{x}) = s] = 0$ as well, and so $P[A(\mathbf{x}) \in S] = 0$, verifying (1.1). Otherwise, if $\mathsf{P}[A(\mathbf{x}') \in S] > 0$,

$$\frac{\mathsf{P}[A(\mathbf{x}) \in S]}{\mathsf{P}[A(\mathbf{x}') \in S]} = \frac{\sum_{s \in S} \mathsf{P}[A(\mathbf{x}) = s]}{\sum_{s \in S} \mathsf{P}[A(\mathbf{x}') = s]} \leq \max_{s \in S} \frac{\mathsf{P}[A(\mathbf{x}) = s]}{\mathsf{P}[A(\mathbf{x}') = s]} \leq \exp(\epsilon),$$

using Lemma 4.1.
($\Rightarrow$) Take $S = \{s\}$ in (1.1). □

2 Bit vector reporting

Our goal is to establish differential privacy properties for user data reported in the form of vectors of bits. To protect user privacy, each user record is randomized prior to leaving the client and anonymized on reaching the server. We now describe the randomization procedure, and place ourselves in the setting of Section 1 by representing it as a query applied to a database.

2.1 Bit randomization

For our purposes, a **bit** is an integer $b \in \{0, 1\}$, and a **bit vector** is a vector $x \in \{0, 1\}^L$. Bits and bit vectors are randomized in the following way.

Definition. The **bit randomization** procedure R with **lie probability** $0 < q < 1/2$ flips a bit b with probability q, and leaves it as-is with probability $p := 1 - q$:

$$R(b) = \begin{cases} b & \text{with prob } p \\ 1 - b & \text{with prob } q \end{cases}.$$

This can be expressed concisely as

$$R(b) = R(b; \xi) = b \cdot \xi + (1 - b) \cdot (1 - \xi) \quad \text{where } \xi \sim Ber(p).$$

We extend the procedure to **bit vector randomization** by applying the randomization independently to each bit in the vector. Given a bit vector $x = (b_1, \ldots, b_L)$, define

$$R(x) = R(x; \xi) = \big(R(b_1; \xi_1), \ldots, R(b_L; \xi_L)\big) \quad \text{where } \xi = (\xi_1, \ldots, \xi_L) \overset{\text{iid}}{\sim} Ber(p).$$

Remark. Note that R reports the original bit value with probability $p = 1 - q > q$, and lies with probability q. This is equivalent to the randomized response procedure where the value is reported as-is with probability $1 - f$, and with probability f the reported value is the outcome of the toss of a fair coin. In our case, $q = f/2$.

Remark. If $q = 1/2$, then $R(0) \overset{d}{=} R(1)$, and the reported value is "completely" randomly generated, i.e., independently of the original value.

We now consider the distribution of the randomized bit vectors. It can be expressed in terms of the Hamming distance between the original and randomized vectors:

$$\delta(x, x') = \sum_{\ell=1}^{L} I(x_\ell \neq x'_\ell) = \sum_{\ell=1}^{L} |x_\ell - x'_\ell|.$$

For a single bit, the randomization has lied when the outcome is different from the original value:

$$\mathsf{P}[R(b) = y] = p^{I(b=y)} \cdot q^{I(b \neq y)} = (1-q)^{1-\delta(b,y)} \cdot q^{\delta(b,y)}.$$

For a bit vector x, this becomes

$$\mathsf{P}[R(x) = y] = p^{\sum I(x_\ell = y_\ell)} \cdot q^{\sum I(x_\ell \neq y_\ell)} = (1-q)^{L-\delta(x,y)} \cdot q^{\delta(x,y)}.$$

Note that this probability is maximized when $\delta(x, y) = 0$ (the randomized vector y is identical to the original vector x), and minimized when $\delta(x, y) = L$. In the latter case, we say that y is the **opposite** of x. In other words, the most likely outcome of randomizing a bit vector is obtaining an identical vector.

2.2 Reporting for bit records

We now place ourselves in the setting of Section 1 for bit-vector user records, as required in the sequel.

Set $\mathcal{D} = \{0, 1\}^L$. We use the term **collection** (of records) interchangeably with "database". We consider a randomized query A that randomizes each record in the collection independently, and aggregates the results by reporting occurrence counts for every possible randomization outcome. We adopt this aggregation step as a model for anonymization. After anonymization, any link to the original collection or ordering is lost, and the information contained in the results is embodied solely by the reported values.

In the following, we rely on the fact that $\mathcal{D}$ is finite, and we assume a specific enumeration $\mathcal{D} = (d_1, \ldots, d_{2^L})$. The ordering is unimportant at this point, although it will be convenient to assume that $d_1 = (1, \ldots, 1)$ and $d_{2^L} = (0, \ldots, 0)$.

Definition. The randomized query $A : \mathcal{D}^n \to \mathcal{S} = \{0, \ldots, n\}^{2^L}$ maps collections of bit vectors to occurrence counts as follows.

(a) Extend the bit randomization R to collections $\mathbf{x}$ by applying it independently to each vector:

$$R(\mathbf{x}) = R(\mathbf{x};\boldsymbol{\xi}) = \big(R(x_1;\xi_1),\dots,R(x_n;\xi_n)\big) \quad \text{where } \xi_i = (\xi_{i1},\dots,\xi_{iL}) \text{ and } \xi_{i\ell} \overset{\text{iid}}{\sim} Ber(p).$$

We call $R(\mathbf{x})$ the **synthetic** collection obtained from the **original** collection $\mathbf{x}$.

(b) Define the function Φ that counts occurrences of the elements of $\mathcal{D} = (d_1,\dots,d_{2^L})$ in a collection $\mathbf{y}$:

$$\Phi(\mathbf{y}) := \left(\sum_{i=1}^{n} I(y_i = d_1),\dots,\sum_{i=1}^{n} I(y_i = d_{2^L}).\right)$$

Note that, if $\Phi(\mathbf{y}) = (s_1,\dots,s_{2^L})$, then $s_1 + \cdots + s_{2^L} = n$.

Finally,

$$A = \Phi \circ R.$$

The distribution of $A(\mathbf{x})$ is given by

$$\begin{aligned}
\mathsf{P}[A(\mathbf{x}) = \mathbf{s}] = \mathsf{P}[\Phi(R(\mathbf{x})) = s] &= \sum_{\mathbf{y}:\Phi(\mathbf{y})=s} \mathsf{P}[R(\mathbf{x}) = \mathbf{y}] \\
&= \sum_{\mathbf{y}:\Phi(\mathbf{y})=s} \prod_{i=1}^{n} \mathsf{P}[R(x_i) = y_i] \qquad (2.1)\\
&= \sum_{\mathbf{y}:\Phi(\mathbf{y})=s} \prod_{i=1}^{n} p^{L-\delta(x_i,y_i)} \cdot q^{\delta(x_i,y_i)} = \sum_{\mathbf{y}:\Phi(\mathbf{y})=s} p^{nL-\delta(\mathbf{x},\mathbf{y})} \cdot q^{\delta(\mathbf{x},\mathbf{y})}
\end{aligned}$$

Note that the support of A is the support of a multinomial random variable with n trials:

$$\mathcal{S}_n := \left\{ s \in \mathcal{S} : \sum_j s_j = n \right\}.$$

3 Maximal collections

We will study the differential privacy of the query A in terms of the **privacy ratio**

$$\pi(\mathbf{s},\mathbf{x},\mathbf{x}') := \frac{\mathsf{P}[A(\mathbf{x}') = \mathbf{s}]}{\mathsf{P}[A(\mathbf{x}) = \mathbf{s}]}$$

for two collections $\mathbf{x},\mathbf{x}' \in \mathcal{D}^n$ differing in one row and $\mathbf{s} \in \mathcal{S}_n$. Note that π is well-defined, since any outcome in $\mathcal{S}_n$ occurs with non-zero probability starting from any collection $\mathbf{x}$. By Proposition 1.1, A is ϵ-differentially private if π is bounded everywhere on $\mathcal{S}$, with $\epsilon = \max_{\mathbf{x},\mathbf{x}'\in\mathcal{D}^n;\,\mathbf{s}\in\mathcal{S}} \log \pi$.

Without loss of generality (in light of (2.1)), assume that the element differing between $\mathbf{x}$ and $\mathbf{x}'$ is the first one, and denote $\tilde{\mathbf{x}} = (x_2,\dots,x_n)$. In other words, $\mathbf{x} = (x_1,\tilde{\mathbf{x}})$ and $\mathbf{x}' = (x_1',\tilde{\mathbf{x}})$. We also write $\mathbf{i}_j = (0,\dots,0,1,0,\dots,0)$, the vector with a 1 in the j-th position and the rest 0.

Conditioning on its value, we can write

$$\pi(\mathbf{s},\mathbf{x},\mathbf{x}') = \frac{\sum_{j=1}^{2^L} \mathsf{P}[A(x_1') = \mathbf{i}_j]\,\mathsf{P}[A(\tilde{\mathbf{x}}) = \mathbf{s} - \mathbf{i}_j]}{\sum_{j=1}^{2^L} \mathsf{P}[A(x_1) = \mathbf{i}_j]\,\mathsf{P}[A(\tilde{\mathbf{x}}) = \mathbf{s} - \mathbf{i}_j]} = \frac{\sum_{j=1}^{2^L} \mathsf{P}[R(x_1') = d_j]\,\mathsf{P}[A(\tilde{\mathbf{x}}) = \mathbf{s} - \mathbf{i}_j]}{\sum_{j=1}^{2^L} \mathsf{P}[R(x_1) = d_j]\,\mathsf{P}[A(\tilde{\mathbf{x}}) = \mathbf{s} - \mathbf{i}_j]}, \tag{3.1}$$

implying that π only depends on the modified collection through the modified record: $\pi = \pi(\mathbf{s},\mathbf{x},x_1')$.

TODO: consequences of this:
- difference between numerator and denominator is basically a reweighting of the same terms in the sum
- max value of ratio

The first question we address is for which pair of original and modified collections $\mathbf{x}$ and $\mathbf{x}'$ does π obtain its maximum.

3.1 The case $L = 1$

It is instructive to first consider the case where each record in the collection consists of a single bit, as the expressions simplify considerably. In this case, the outcome of A essentially reduces to the number of 1s obtained in the synthetic collection $R(\mathbf{x})$, since $\mathbf{s} \in \mathcal{S}_n$ can be written as (s_1, s_2) with $s_1 \in \{0,\dots,n\}$ and $s_2 = n - s_1$. Using this fact, we interpret $A(\mathbf{x})$ as $\sum_{i=1}^n I(R(x_i) = 1)$, and express the privacy ratio as

$$\pi(s,\mathbf{x},x_1') = \frac{\mathsf{P}[R(x_1') = 1]\,\mathsf{P}[A(\tilde{\mathbf{x}}) = s-1] + \mathsf{P}[R(x_1') = 0]\,\mathsf{P}[A(\tilde{\mathbf{x}}) = s]}{\mathsf{P}[R(x_1) = 1]\,\mathsf{P}[A(\tilde{\mathbf{x}}) = s-1] + \mathsf{P}[R(x_1) = 0]\,\mathsf{P}[A(\tilde{\mathbf{x}}) = s]}$$

for $s \in \{0,\dots,n\}$. Furthermore, the requirement that $x_1 \neq x_1'$ in the single-bit case implies that $x_1' = 1 - x_1$, so

$$\pi(s,\mathbf{x},x_1') = \frac{(1-\mathsf{P}[R(x_1) = 1])\,\mathsf{P}[A(\tilde{\mathbf{x}}) = s-1] + (1-\mathsf{P}[R(x_1) = 0])\,\mathsf{P}[A(\tilde{\mathbf{x}}) = s]}{\mathsf{P}[R(x_1) = 1]\,\mathsf{P}[A(\tilde{\mathbf{x}}) = s-1] + \mathsf{P}[R(x_1) = 0]\,\mathsf{P}[A(\tilde{\mathbf{x}}) = s]}.$$

We fix $s \in \{1,\dots,n\}$ and investigate which choice of collection $\mathbf{x} = (x_1,\tilde{\mathbf{x}})$ maximizes $\pi(s,\mathbf{x},x_1')$. Assume first $x_1 = 1$. (TODO: what about when $x_1 = 0$?) Then

$$\pi(s,\tilde{\mathbf{x}},0) = \frac{q\,\mathsf{P}[A(\tilde{\mathbf{x}}) = s-1] + p\,\mathsf{P}[A(\tilde{\mathbf{x}}) = s]}{p\,\mathsf{P}[A(\tilde{\mathbf{x}}) = s-1] + q\,\mathsf{P}[A(\tilde{\mathbf{x}}) = s]}.$$

By Lemma 4.2, this ratio is maximized at $\tilde{\mathbf{x}}^*$ satisfying

$$\frac{\mathsf{P}[A(\tilde{\mathbf{x}}^*) = s]}{\mathsf{P}[A(\tilde{\mathbf{x}}^*) = s-1]} \geq \frac{\mathsf{P}[A(\tilde{\mathbf{y}}) = s]}{\mathsf{P}[A(\tilde{\mathbf{y}}) = s-1]}$$

for any $\tilde{\mathbf{y}} \in \mathcal{D}^{n-1}$. We claim that this is the case when $\tilde{\mathbf{x}}^*$ consists of all 1s:

Claim. *Write* $\mathbf{1} = (1, ..., 1)$ *as an element of* $\mathcal{D}^n$. *Then, given* $s \in \{1, \ldots, n\}$,

$$\frac{\mathsf{P}[A(\mathbf{1}) = s]}{\mathsf{P}[A(\mathbf{1}) = s-1]} \geq \frac{\mathsf{P}[A(\mathbf{y}) = s]}{\mathsf{P}[A(\mathbf{y}) = s-1]} \tag{3.2}$$

for any $\mathbf{y} \in \mathcal{D}^n$.

Proof. We proceed by induction on n. If $n = 1$, then $A(x) = R(x)$, we need only confirm (3.2) for $s = 1$. Taking $\mathbf{y} = 0$ (the only possibility aside from $\mathbf{1}$), we have

$$\frac{\mathsf{P}[R(0) = 1]}{\mathsf{P}[R(0) = 0]} = \frac{q}{p} \leq \frac{p}{q} = \frac{\mathsf{P}[R(1) = 1]}{\mathsf{P}[R(1) = 0]} \tag{3.3}$$

verifying (3.2) in this case. Next, suppose that (3.2) holds for $s \in \{1, \ldots, n-1\}$ and $\mathbf{1}, \tilde{\mathbf{y}} \in \mathcal{D}^{n-1}$, and consider $\mathbf{y} = (\tilde{\mathbf{y}}, y_n) \in \mathcal{D}^n$. For convenience, write $p(y_n) = \mathsf{P}[R(y_n) = 1]$. Observe that

$$\mathsf{P}[A(\mathbf{y}) = s] = \mathsf{P}[A(\tilde{\mathbf{y}}) = s] \cdot (1 - p(y_n)) + \mathsf{P}[A(\tilde{\mathbf{y}}) = s-1] \cdot p(y_n),$$

conditioning on the value of y_n. Thus,

$$\frac{\mathsf{P}[A(\mathbf{y}) = s]}{\mathsf{P}[A(\mathbf{y}) = s-1]} = \frac{\mathsf{P}[A(\tilde{\mathbf{y}}) = s] \cdot (1 - p(y_n)) + \mathsf{P}[A(\tilde{\mathbf{y}}) = s-1] \cdot p(y_n)}{\mathsf{P}[A(\tilde{\mathbf{y}}) = s-1] \cdot (1 - p(y_n)) + \mathsf{P}[A(\tilde{\mathbf{y}}) = s-2] \cdot p(y_n)}. \tag{3.4}$$

For $s \in \{2, \ldots, n-1\}$, our induction hypothesis implies that

$$\frac{\mathsf{P}[A(\tilde{\mathbf{y}}) = s]}{\mathsf{P}[A(\tilde{\mathbf{y}}) = s-1]} \leq \frac{\mathsf{P}[A(\mathbf{1}) = s]}{\mathsf{P}[A(\mathbf{1}) = s-1]} \quad \text{and} \quad \frac{\mathsf{P}[A(\tilde{\mathbf{y}}) = s-1]}{\mathsf{P}[A(\tilde{\mathbf{y}}) = s-2]} \leq \frac{\mathsf{P}[A(\mathbf{1}) = s-1]}{\mathsf{P}[A(\mathbf{1}) = s-2]},$$

and furthermore,

$$\frac{\mathsf{P}[A(\tilde{\mathbf{y}}) = s]}{\mathsf{P}[A(\tilde{\mathbf{y}}) = s-2]} = \frac{\mathsf{P}[A(\tilde{\mathbf{y}}) = s]}{\mathsf{P}[A(\tilde{\mathbf{y}}) = s-1]} \cdot \frac{\mathsf{P}[A(\tilde{\mathbf{y}}) = s-1]}{\mathsf{P}[A(\tilde{\mathbf{y}}) = s-2]} \leq \frac{\mathsf{P}[A(\mathbf{1}) = s]}{\mathsf{P}[A(\mathbf{1}) = s-2]}.$$

Therefore, we can apply Lemma 4.3 to (3.4) to obtain

$$\frac{\mathsf{P}[A(\mathbf{y}) = s]}{\mathsf{P}[A(\mathbf{y}) = s-1]} \leq \frac{\mathsf{P}[A(\mathbf{1}) = s] \cdot (1 - p(y_n)) + \mathsf{P}[A(\mathbf{1}) = s-1] \cdot p(y_n)}{\mathsf{P}[A(\mathbf{1}) = s-1] \cdot (1 - p(y_n)) + \mathsf{P}[A(\mathbf{1}) = s-2] \cdot p(y_n)} \tag{3.5}$$

for $s \in \{2, \ldots, n-1\}$. If $s = n$, we consider $\mathsf{P}[A(\tilde{\mathbf{y}}) = s] = 0$ since $\mathsf{P}[A(\tilde{\mathbf{y}}) \in \{0, \ldots, n-1\}] = 1$, and similarly for $\mathsf{P}[A(\mathbf{1}) = s]$. In this case, Lemma 4.3 still applies with $b = b' = 0$. A similar argument establishes (3.5) when $s = 1$. Therefore,

$$\frac{\mathsf{P}[A(\mathbf{y}) = s]}{\mathsf{P}[A(\mathbf{y}) = s-1]} \leq \frac{\mathsf{P}[A((\mathbf{1}, y_n)) = s]}{\mathsf{P}[A((\mathbf{1}, y_n)) = s-1]}$$

for $s \in \{1, \ldots, n\}$. Since $y_n \in \{0, 1\}$, the proof will be complete if we show that

$$\begin{aligned}\frac{\mathsf{P}[A((\mathbf{1}, 1)) = s]}{\mathsf{P}[A((\mathbf{1}, 1)) = s-1]} &= \frac{\mathsf{P}[A(\mathbf{1}) = s] \cdot q + \mathsf{P}[A(\mathbf{1}) = s-1] \cdot p}{\mathsf{P}[A(\mathbf{1}) = s-1] \cdot q + \mathsf{P}[A(\mathbf{1}) = s-2] \cdot p} \\ &\geq \frac{\mathsf{P}[A(\mathbf{1}) = s] \cdot p + \mathsf{P}[A(\mathbf{1}) = s-1] \cdot q}{\mathsf{P}[A(\mathbf{1}) = s-1] \cdot p + \mathsf{P}[A(\mathbf{1}) = s-2] \cdot q} = \frac{\mathsf{P}[A((\mathbf{1}, 0)) = s]}{\mathsf{P}[A((\mathbf{1}, 0)) = s-1]}.\end{aligned} \tag{3.6}$$

Using the fact that $A(\mathbf{1}) \sim Bin(n-1, p)$, observe that

$$\frac{\mathsf{P}[A(\mathbf{1}) = s]}{\mathsf{P}[A(\mathbf{1}) = s-1]} = \frac{\binom{n-1}{s} p^s q^{n-1-s}}{\binom{n-1}{s-1} p^{s-1} q^{n-s}} = \frac{n-s}{s}\frac{p}{q},$$

from which

$$\frac{\lambda_2}{\mu_2} = \frac{\mathsf{P}[A(\mathbf{1}) = s]}{\mathsf{P}[A(\mathbf{1}) = s-1]} = \frac{n-s}{s}\frac{p}{q} \leq \frac{n-s+1}{s-1}\frac{p}{q} = \frac{\mathsf{P}[A(\mathbf{1}) = s-1]}{\mathsf{P}[A(\mathbf{1}) = s-2]} = \frac{\lambda_1}{\mu_1},$$

and hence (3.6) follows from Lemma 4.2 (unless $s = 1$, in which case it follows from a simple direct argument). □

TODO: finish argument in the case when $x_1 = 0$.

4 Ratios of sums: properties

Here we establish some results around bounding and comparing ratios of sums, which will be useful in working with the privacy ratio.

Lemma 4.1. *Suppose $a_1, \ldots, a_m, b_1, \ldots, b_m \in \mathbb{R}$ with $b_i > 0$ all i. Then*

$$\frac{a_1 + \cdots + a_m}{b_1 + \cdots + b_m} \leq \max\left(\frac{a_1}{b_1}, \ldots, \frac{a_m}{b_m}\right).$$

Proof. Write

$$\frac{a_1 + \cdots + a_m}{b_1 + \cdots + b_m} = \frac{a_1}{b_1}\frac{b_1}{b_1 + \cdots + b_m} + \cdots + \frac{a_m}{b_m}\frac{b_m}{b_1 + \cdots + b_m} = \sum_{i=1}^{m} \frac{a_i}{b_i}\lambda_i$$

where $\lambda_1 + \cdots + \lambda_m = 1$. The result follows since each a_i/b_i is bounded by $\max_i a_i/b_i$. □

Lemma 4.2. *Suppose $a_i, a'_i, \lambda_i, \mu_i > 0$ for $i = 1, \ldots, m$. Then*

$$\frac{a_1\lambda_1 + \cdots + a_m\lambda_m}{a_1\mu_1 + \cdots + a_m\mu_m} \geq \frac{a'_1\lambda_1 + \cdots + a'_m\lambda_m}{a'_1\mu_1 + \cdots + a'_m\mu_m} \tag{4.1}$$

if

$$\lambda_i/\mu_i \geq \lambda_j/\mu_j \tag{4.2}$$

and

$$a_i/a_j \geq a'_i/a'_j \tag{4.3}$$

whenever $1 \leq i < j \leq m$.

Note that numerator and denominator have the same index (4.2), and different indices in (4.3). It is easy to see that (4.2) is satisfied when $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ and $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_m$. It is also clear from the proof that (4.1) still holds if all the inequalities in (4.2) and (4.3) are reversed.

Proof. Cross-multiplying, we see that (4.1) is equivalent to

$$\sum_i \sum_j a_i a'_j \lambda_i \mu_j \geq \sum_i \sum_j a_i a'_j \lambda_j \mu_i \iff \sum_i \sum_{j \neq i} a_i a'_j (\lambda_i \mu_j - \lambda_j \mu_i) \geq 0.$$

Furthermore,

$$\begin{aligned}
\sum_i \sum_{j \neq i} a_i a'_j (\lambda_i \mu_j - \lambda_j \mu_i) &= \sum_i \sum_{j>i} a_i a'_j (\lambda_i \mu_j - \lambda_j \mu_i) + \sum_i \sum_{j<i} a_i a'_j (\lambda_i \mu_j - \lambda_j \mu_i) \\
&= \sum_i \sum_{j>i} a_i a'_j (\lambda_i \mu_j - \lambda_j \mu_i) + \sum_j \sum_{i<j} a_j a'_i (\lambda_j \mu_i - \lambda_i \mu_j) \\
&= \sum_i \sum_{j>i} a_i a'_j (\lambda_i \mu_j - \lambda_j \mu_i) + \sum_i \sum_{j>i} a_j a'_i (\lambda_j \mu_i - \lambda_i \mu_j) \\
&= \sum_i \sum_{j>i} (a_i a'_j - a_j a'_i)(\lambda_i \mu_j - \lambda_j \mu_i),
\end{aligned}$$

where the second equality follows from relabeling the summation indices, and the third from reversing the sums. It follows that (4.1) will hold if $(a_i a'_j - a_j a'_i)(\lambda_i \mu_j - \lambda_j \mu_i) \geq 0$ for all $1 \leq i < j \leq m$, which is implied by (4.2) and (4.3). □

Lemma 4.3. *Suppose $a, a', \lambda, \mu > 0$, and $b, b', c, c' \geq 0$. Then*

$$\frac{a\lambda + b\mu}{c\lambda + a\mu} \geq \frac{a'\lambda + b'\mu}{c'\lambda + a'\mu} \tag{4.4}$$

if

$$ac' \geq a'c, \quad ab' \leq a'b, \quad \text{and} \quad bc' \geq b'c. \tag{4.5}$$

Proof. (4.4) holds iff

$$\begin{aligned}
& ac'\lambda^2 + aa'\lambda\mu + bc'\lambda\mu + a'b\mu^2 \geq a'c\lambda^2 + b'c\lambda\mu + aa'\lambda\mu + ab'\mu^2 \\
\iff & (ac' - a'c)\lambda^2 + (bc' - b'c)\lambda\mu + (a'b - ab')\mu^2 \geq 0,
\end{aligned}$$

which is implied by (4.5). □

5 Old stuff

Furthermore, if A randomizes each record in the database independently, i.e., $A(\boldsymbol{x}) = A(\boldsymbol{x}, \boldsymbol{X}) := \big(A_0(x_1, X_1), \ldots, A_0(x_n, X_n)\big)$ where X_i are independent, then $\boldsymbol{S} = \boldsymbol{S}_0^n$ and $s = (s_1, \ldots, s_n)$ with $s_i \in \boldsymbol{S}_0$. In this case $P[A(\boldsymbol{x}) = s] = P[A_0(x_1) = s_1, \ldots, A_0(x_n) = s_n] = \prod P[A_0(x_i) = s_i]$. If $\boldsymbol{x}$ and $\boldsymbol{x'}$ differ in one row (wlog $x_1 \neq x'_1$ and $x_i = x'_i$ for $i = 2, \ldots, n$), then

$$\frac{P[A(\boldsymbol{x}) = s]}{P[A(\boldsymbol{x'}) = s]} = \frac{P[A_0(x_1) = s_1]}{P[A_0(x'_1) = s_1]}.$$

Therefore, in this case, the query A will satisfy differential privacy if

$$P[A_0(x) = s] \leq \epsilon \cdot P[A_0(x') = s]$$

for all $x, x' \in D$ and $s \in \boldsymbol{S}_0$. This is the formulation used in the RAPPOR paper that applies to differences between individual records rather than collections differing on a single element.