K-Randomization

Maxim Zhilyaev David Zeber

October 23, 2015

Outline of the procedure

-

1 Theoretical setup

In the following we work with data in the form of bit vectors. A **bit vector** is a vector $v \in \{0,1\}^L$.

First we define the randomization procedure we will be applying.

Definition. The randomization procedure R with **lie probability** $0 < q < 1/2$ flips a bit with probability q, and leaves it as-is with probability $1-q$. In other words, for a bit $b \in \{0,1\}$,

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

When applied to a vector, each bit is randomized independently:

$$R(v) = R\big(v; (X_1, \ldots, X_L)\big) = \big(R(v_1; X_1), \ldots, R(v_L; X_L)\big) \quad \text{where } X \overset{\text{iid}}{\sim} Ber(q).$$

Remark. The randomization R reports the original bit value with probability $1-q > q$, 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 this 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.

Distribution of $R(v)$.

For a bit b, the randomization lies iff $R(b) \neq b$:

$$P[R(b) = s] = q^{\mathbf{1}_{\{b \neq s\}}} (1-q)^{\mathbf{1}_{\{b = s\}}}$$

Hence, for a bit vector v,

$$P[R(v) = s] = q^{\sum \mathbf{1}_{\{b_i \neq s_i\}}}(1-q)^{\sum \mathbf{1}_{\{b_i = s_i\}}} = q^{L-m(v,s)}(1-q)^{m(v,s)},$$

where $m(v,s) = |\{i : v_i = s_i\}|$. Note that this probability is maximized when $m(v,s) = L$ (the reported vector s is identical to the original vector v), and minimized when $m(v,s) = 0$. In other words, the most likely outcome of randomizing a bit vector is obtaining an identical vector.

For a collection T,

$$P[s \in R(T)] = 1 - P[s \notin R(T)] = 1 - \prod_{v \in T} P[R(v) \neq s] = 1 - \prod_{v \in T} \left[1 - q^{L-m(v,s)}(1-q)^{m(v,s)}\right].$$

2 Differential Privacy

The typical setting for differential privacy is the following. We consider a **database** as a collection of records. The records are elements of some space D, and a database $\boldsymbol{x}$ is a vector of n records: $\boldsymbol{x} \in D^n$.

We wish to release information based on the database by applying a **query** to it. This is a function A mapping the database into another space: $A : D^n \to \boldsymbol{S}$. If the function A is random, i.e., $A(\boldsymbol{x}) = A(\boldsymbol{x}, X)$ for a random element X, then the output $A(\boldsymbol{x})$ is a random element of $\boldsymbol{S}$.

In considering the differential privacy of A, we compare the result of applying A to two very similar databases $\boldsymbol{x}$, $\boldsymbol{x'} \in D^n$. We say the databases **differ in one row** if $\sum_{i=1}^n \mathbf{1}_{\{x_i \neq x_i'\}} = 1$. The random query A is said to be ϵ-**differentially private** if, for any two databases $\boldsymbol{x}$, $\boldsymbol{x'} \in D^n$ differing in one row,

$$P[A(\boldsymbol{x}) \in S] \leq \epsilon \cdot P[A(\boldsymbol{x'}) \in S]$$

for all $S \subset \boldsymbol{S}$ (measurable). An alternative notion of differing in one row that is sometimes used is that $\boldsymbol{x} \in D^n$, $\boldsymbol{x'} \in D^{n+1}$, and $x_i = x_i'$ for $i = 1, \ldots, n$. In other words, $\boldsymbol{x'}$ includes an additional record that is not in $\boldsymbol{x}$.

If $\boldsymbol{S}$ is countable, then we can write

$$P[A(\boldsymbol{x}) \in S] = \sum_{s \in S} P[A(\boldsymbol{x}) = s].$$

Hence,

$$\frac{P[A(\boldsymbol{x}) \in S]}{P[A(\boldsymbol{x'}) \in S]} = \frac{\sum_{s \in S} P[A(\boldsymbol{x}) = s]}{\sum_{s \in S} P[A(\boldsymbol{x'}) = s]} \leq \max_{s \in S} \frac{P[A(\boldsymbol{x}) = s]}{P[A(\boldsymbol{x'}) = s]}$$

by the Lemma (need reference).

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.

Consider a collection T of bit vectors, and write $T_v = T \backslash \{v\}$. The randomization procedure R is ϵ-differentially private if

$$\log \left(\frac{P[R(T) \in S]}{P[R(T_v) \in S]} \right) \leq \epsilon$$

for any set of bit vectors S.

Anonymity:

$$A_p = \min_{v \in T, s \in \{0,1\}^L} \frac{P[s \in R(T_v)]}{P[s = R(v)]}$$