RUB

RUHR-UNIVERSITÄT BOCHUM

Description Length Bounds II

Oberseminar über Differential Privacy, 21th Mai 2019

Alexander Helm

Overview

1. Properties of Transcript Compressibility
   - Postprocessing
   - Composition

2. Building Estimators
   - Trivial Estimator
   - Ladder Mechanism

Class of Queries

Definition (Statistical Query)

Given some data domain $\mathcal{X}$, a statistical query ϕ is a function $\phi : \mathcal{X} \to [0, 1]$.

Class of Queries

Definition (Statistical Query)

Given some data domain $\mathcal{X}$, a statistical query ϕ is a function $\phi : \mathcal{X} \to [0, 1]$.

Definition (c-sensitive Query)

A query $q : \mathcal{X}^n \to \mathbb{R}$ has *sensitivity* c if for all $x_1, x_2, \ldots, x_n \in \mathcal{X}$, all indices i, and all $x_i' \in \mathcal{X}$:

$$|q(x_1, \ldots, x_n) - q(x_1, \ldots, x_{i-1}, x_i', x_{i+1} \ldots, x_n)| \leq c$$

Class of Queries

Definition (Statistical Query)

Given some data domain $\mathcal{X}$, a statistical query ϕ is a function $\phi : \mathcal{X} \to [0, 1]$.

Definition (c-sensitive Query)

A query $q : \mathcal{X}^n \to \mathbb{R}$ has *sensitivity* c if for all $x_1, x_2, \ldots, x_n \in \mathcal{X}$, all indices i, and all $x_i' \in \mathcal{X}$:

$$|q(x_1, \ldots, x_n) - q(x_1, \ldots, x_{i-1}, x_i', x_{i+1} \ldots, x_n)| \leq c$$

- statistical queries are c-sensitive for $c = 1/n$

Class of Queries

Definition (Statistical Query)

Given some data domain $\mathcal{X}$, a statistical query ϕ is a function $\phi : \mathcal{X} \to [0, 1]$.

Definition (c-sensitive Query)

A query $q : \mathcal{X}^n \to \mathbb{R}$ has *sensitivity* c if for all $x_1, x_2, \ldots, x_n \in \mathcal{X}$, all indices i, and all $x_i' \in \mathcal{X}$:

$$|q(x_1, \ldots, x_n) - q(x_1, \ldots, x_{i-1}, x_i', x_{i+1} \ldots, x_n)| \leq c$$

- statistical queries are c-sensitive for $c = 1/n$
- Generalized Transcript Compressibility Transfer Theorem for $1/n$-sensitive queries

Postprocessing for Transcript Compressibility

GenerateTranscript$_{n,k}(\mathcal{A}, S, f \circ \mathcal{O}, \mathcal{Q})$

S is given to $\mathcal{O}$.

for $i = 1$ to k **do**

$\mathcal{A}$ chooses a query $q_i \in Q$. $\hat{q}_i = f(q_i)$ is given to $\mathcal{O}$.

$\mathcal{O}$ generates an answer $a_i \in [0, 1]$. $\hat{a}_i = f(a_i)$ is given to $\mathcal{A}$.

end for

The *transcript* $T = (\hat{q}_1, \hat{a}_1, \ldots, \hat{q}_k, \hat{a}_k)$ is output

Postprocessing for Transcript Compressibility

GenerateTranscript$_{n,k}(\mathcal{A}, S, f \circ \mathcal{O}, \mathcal{Q})$

S is given to $\mathcal{O}$.

for $i = 1$ to k **do**

$\mathcal{A}$ chooses a query $q_i \in Q$. $\hat{q}_i = f(q_i)$ is given to $\mathcal{O}$.

$\mathcal{O}$ generates an answer $a_i \in [0,1]$. $\hat{a}_i = f(a_i)$ is given to $\mathcal{A}$.

end for

The *transcript* $T = (\hat{q}_1, \hat{a}_1, \ldots, \hat{q}_k, \hat{a}_k)$ is output

Theorem (Postprocessing)

Suppose $\mathcal{O} : \mathcal{Q} \to \mathcal{R}$ is b-transcript compressible. Let be $f : \mathcal{Q} \cup \mathcal{R} \to \mathcal{Q} \cup \mathcal{R}$ an arbitrary stateful algorithm. Then $(f \circ \mathcal{O})$ is also b-transcript compressible.

Composition for Transcript Compressibility

GenerateTranscript$_{n,k_1+k_2}(\mathcal{A}, S, (\mathcal{O}_1, \mathcal{O}_2), \mathcal{Q})$

S is given to $\mathcal{O}$.
for $i = 1$ to k_1 **do**
  $\mathcal{A}$ chooses a query $q_i \in Q$. q_i is given to $\mathcal{O}_1$.
  $\mathcal{O}_1$ generates an answer $a_i \in [0, 1]$. a_i is given to $\mathcal{A}$.
end for
for $i = k_1 + 1$ to $k_1 + k_2$ **do**
  $\mathcal{A}$ chooses a query $q_i \in Q$. q_i is given to $\mathcal{O}_2$.
  $\mathcal{O}_2$ generates an answer $a_i \in [0, 1]$. a_i is given to $\mathcal{A}$.
end for
The *transcript* $T = (q_1, a_1, \ldots, q_{k_1+k_2}, a_{k_1+k_2})$ is output

Composition for Transcript Compressibility

Theorem (Composition)

Suppose $\mathcal{O}_1 : \mathcal{Q} \to \mathcal{R}$ *is transcript compressible to* $b_1(n, k_1)$ *bits, and* $\mathcal{O}_2 : \mathcal{Q} \to \mathcal{R}$ *is transcript compressible to* $b_2(n, k_2)$ *bits. Then the composition* $(\mathcal{O}_1, \mathcal{O}_2)$ *is transcript compressible to* $b(n, k_1 + k_2) = b_1(n, k_1) + b_2(n, k_2)$ *bits.*

Trivial estimator

Definition (b-bit truncated estimator)

Given a dataset S, the b-bit truncated estimator $\mathcal{O}_b^T(q)$ returns $q(S)$ truncated to b bits of binary precision.

Trivial estimator

Definition (b-bit truncated estimator)

Given a dataset S, the b-bit truncated estimator $\mathcal{O}_b^T(q)$ returns $q(S)$ truncated to b bits of binary precision.

- on a single query $\mathcal{O}_b^T$ is transcript compressible to b bits

Trivial estimator

Definition (b-bit truncated estimator)

Given a dataset S, the b-bit truncated estimator $\mathcal{O}_b^T(q)$ returns $q(S)$ truncated to b bits of binary precision.

- on a single query $\mathcal{O}_b^T$ is transcript compressible to b bits
- for k queries:
  $\Rightarrow b(n,k)$-transcript compressible for $b(n,k) = b \cdot k$.

Trivial estimator

Definition (b-bit truncated estimator)

Given a dataset S, the b-bit truncated estimator $\mathcal{O}_b^T(q)$ returns $q(S)$ truncated to b bits of binary precision.

- on a single query $\mathcal{O}_b^T$ is transcript compressible to b bits
- for k queries:
  $\Rightarrow b(n,k)$-transcript compressible for $b(n,k) = b \cdot k$.
- $\mathcal{O}_b^T$ is $(1/2^b, 0)$-sample accurate

Trivial estimator

Theorem (Accuracy of the b-bit truncated estimator)

Fix any $k < n$ and $\delta > 0$. When $b = \log\sqrt{\frac{n}{k}}$, the b-bit truncated estimator $\mathcal{O}_b^T$ is (ϵ, δ)-accurate for k $1/n$-sensitive queries, where

$$\epsilon = \sqrt{\frac{k}{n}} + \sqrt{\frac{(k \cdot \log\sqrt{\frac{n}{k}} + 1)ln(2) + ln(k/\delta)}{2n}} = \tilde{O}\left(\sqrt{\frac{k + ln(1/\delta)}{n}}\right)$$

Subroutine AboveThreshold

AboveThreshold$(T, q_1, q_2, \ldots)$:

```
AllDone ← FALSE
while not AllDone do
    Accept the next query q_i
    Compute a_i ← q_i(S)
    if a_i < T then
        Return ⊥
    else
        Return ⊤
        AllDone ← TRUE.
    end if
end while
```

Subroutine AboveThreshold

AboveThreshold$(T, q_1, q_2, \ldots)$:

```
AllDone ← FALSE
while not AllDone do
    Accept the next query q_i
    Compute a_i ← q_i(S)
    if a_i < T then
        Return ⊥
    else
        Return ⊤
        AllDone ← TRUE.
    end if
end while
```

Lemma (Compressibility of AboveThreshold)

*For any Threshold T, **AboveThreshold**(T) is transcript compressible to $b(n,k)$ bits, where $b(n,k) = \log(k+1)$.*

Ladder Mechanism

Ladder$(\eta, f_1, f_2, \ldots)$:

Output $\text{BestAccuracy}_0 \leftarrow 0$
for $m = 1$ to $1/\eta$ **do**
  Start an instance of **AboveThreshold** with threshold $T_m = \text{BestAccuracy} + \eta$.
  while AboveThreshold has not halted **do**
    Accept the next classifier f_i.
    Feed **AboveThreshold** the query $q_i(S) = 1 - \ell(f_i(x), y)$.
    if AboveThreshold returns $\perp$ **then**
      Output $\text{BestAccuracy}_i \leftarrow \text{BestAccuracy}_{i-1}$
    end if
  end while
  Output $\text{BestAccuracy}_i = \mathcal{O}_b^T(q_i)$ for $b = \log(1/\eta)$.
end for

- leader query is a $1/n$-sensitive query

Ladder Mechanism

Theorem

Setting $\eta = \left(\frac{\log(k/\delta)}{n}\right)^{1/3}$*, for any* $\delta > 0$*,* ***Ladder*** *is* (ϵ, δ)*-accurate for any set of k leader queries, where*

$$\epsilon = O\left(\left(\frac{\log(k/\delta)}{n}\right)^{1/3}\right)$$