Lecture 4 SVT, Linear Query Release (Part II) and Private Selection

Yu-Xiang Wang

COMPUTER SCIENCE
UC SANTA BARBARA
Computing. ReInvented.

Recap: last lecture

- Generalizing the problem of linear query release
- Apply Laplace mechanism to this problem
  - Releasing queries
  - Releasing data (i.e., contingency table)
- Sparse vector technique
  - Privately selecting an sparse number of queries that are interesting among possibly infinitely many queries

Recap: (Generalized) AboveThreshold mechanism

Algorithm 1 Input is a private database D, an adaptively chosen stream of sensitivity 1 queries $f_1, \ldots$, and a threshold T. Output is a stream of responses $a_1, \ldots$

AboveThreshold$(D, \{f_i\}, T, \epsilon)$

Let $\hat{T} = T + \text{Lap}\left(\frac{2}{\epsilon}\right)$.
for Each query i **do**
  Let $\nu_i = \text{Lap}(\frac{4}{\epsilon})$
  if $f_i(D) + \nu_i \geq \hat{T}$ **then**
    Output $a_i = \top$.
    Halt.
  else
    Output $a_i = \perp$.
  end if
end for

Any noise-adding mechanism M_1 satisfying ε/2-DP for queries with sensitivity 1.

Any noise-adding mechanism M_2 satisfying ε/2-DP for queries with sensitivity 2.

Recap: SparseVector mechanism

1. Start with a budget of ε, and a maximum number of "discoveries" c
2. Split ε into c equal parts and run **AboveThreshold** for up to c times, each with a privacy budget of ε/c.
3. Stop when either the stream of queries are exhausted or if all c discoveries are made.

Recap: the analysis of AboveThreshold

1. The output space of the algorithm

$$\{\perp^k \top | k = 0, 1, ..., \infty\}$$

   - The output can be completely described by a random integer k
2. w.l.o.g., we can assume T = 0 (why?)
3. The probability of outputting k is

$$\begin{aligned}
&\Pr[\mathcal{M}(D) = k] = \mathbb{E}_{z\sim p_\rho}[\Pr[\mathcal{M}(D) = k|z]] \\
=&\mathbb{E}_{z\sim p_\rho}[\prod_{i\le k} \Pr[q_i(D) + \nu_i < z|z]\Pr[q_{k+1}(D) + \nu_i \ge z|z]] \\
=&\int_{-\infty}^{+\infty} p_\rho(z)\left(\prod_{i\le k}\int_{-\infty}^{z-q_i(D)} p(\nu_i)d\nu_i\right)\cdot\int_{z-q_{k+1}(D)}^{\infty} p(\nu_{k+1})d\nu_{k+1}dz
\end{aligned}$$

Recap: The analysis of AboveThreshold

$$\Pr[\mathcal{M}(D) = k] = \mathbb{E}_{z\sim p_\rho}[\Pr[\mathcal{M}(D) = k|z]]$$

$$=\mathbb{E}_{z\sim p_\rho}[\prod_{i\le k} \Pr[q_i(D) + \nu_i < z|z]\Pr[q_{k+1}(D) + \nu_i \ge z|z]]$$

$$= \int_{-\infty}^{+\infty} p_\rho(z)\left(\prod_{i\le k}\int_{-\infty}^{z-q_i(D)} p(\nu_i)d\nu_i\right)\cdot\int_{z-q_{k+1}(D)}^{\infty} p(\nu_{k+1})d\nu_{k+1}dz$$

Key trick: a change of variable that shifts noisy-threshold by Δ

$$\overset{u:=z+\triangle}{\downarrow} = \int_{-\infty}^{+\infty} p_\rho(u-\triangle)\left(\prod_{i\le k}\int_{-\infty}^{u-\triangle-q_i(D)} p(\nu_i)d\nu_i\right)\cdot\int_{u-\triangle-q_{k+1}(D)}^{\infty} p(\nu_{k+1})d\nu_{k+1}du$$

$$= \int_{-\infty}^{+\infty} p_\rho(u)\left(\frac{p_\rho(u-\triangle)}{p_\rho(u)}\right)\left(\prod_{i\le k}\int_{-\infty}^{u-\triangle-q_i(D)} p(\nu_i)d\nu_i\right)\cdot\int_{u-\triangle-q_{k+1}(D)}^{\infty} p(\nu_{k+1})d\nu_{k+1}du$$

$$=\mathbb{E}_{z\sim p_\rho}\left[\left(\frac{p_\rho(z-\triangle)}{p_\rho(z)}\right)\left(\prod_{i\le k}\int_{-\infty}^{z-\triangle-q_i(D)} p(\nu_i)d\nu_i\right)\cdot\int_{z-\triangle-q_{k+1}(D)}^{\infty} p(\nu_{k+1})d\nu_{k+1}\right]$$

Recap: Bounding the third term via a fictitious query

$$\int_{z-\triangle-q_{k+1}(D)}^{\infty} p(\nu_{k+1})d\nu_{k+1} \quad \overset{?}{\longleftrightarrow} \quad \int_{z-q_{k+1}(D')}^{\infty} p(\nu_{k+1})d\nu_{k+1}$$

- Define: $\tilde{q}(\tilde{D}) = \begin{cases} q_{k+1}(D) + \Delta, & \text{if } \tilde{D} = D \\ q_{k+1}(\tilde{D}), & \text{otherwise.} \end{cases}$
- What is the sensitivity of $\tilde{q}$?

This lecture

- Finish the topic on SVT
- Apply SVT for answering many linear queries
  - Private-Multiplicative Weight
- Differentially private selection
  - Exponential mechanism
  - Report Noisy Max

Readings

- For private multiplicative weights algorithm
  - Dwork and Roth, Section 4.2.
  - Alternatively, Vadhan: Section 4.2

- For a proof of the multiplicative weight / hedge
  - Online Convex Optimization book (Hazan), Section 1.3
  - Or watch my video from Convex Optimization (taught in 2020 Spring, will post the link on Piazza)

- For private selection, read:
  - Dwork and Roth Section 3.3 and 3.4
  - **[Advanced reading]** Dong's blog on exponential mechanism https://dongjs.github.io/2020/02/10/ExpMech.html

Utility of SparseVector

- Idea is to simply bound the magnitude of the noise
  - (All true discoveries.) With high probability, we do not wrongly reject interesting queries.
  - (No-false discovery). With high probability, we also do not wrongly identify queries that are not interesting as interesting.
  - Union bound over all of them.

We are outputting only the selections, but not numerical values? NumericSparse!

- This is trivial to fix with twice the privacy budget.
- Compose the following:
  - AboveThresh1, LapMech1, AboveThresh2, LapMech2,…
- Each one of the mechanism is adaptively chosen based on realized previous outcomes.
  - How does LapMech_j depend on the output of AboveThresh_j?
  - How does the AboveThresh_j depend on all previous outputs?

Let's apply the above SVT method for online query release.

- Problem setup:
  - A adaptive online sequence of linear queries.
  - The curator has to answer them as they arrive.

- Baseline:
  - Laplace mechanism for releasing queries $O(|Q|/ \varepsilon)$
  - Laplace mechanism for releasing contingency table $O(\sqrt{|X|} \log |Q| / \varepsilon)$.

- Question: Is it possible to get $O(\text{polylog}(|Q|,|X|))$ error?

Idea: Use correlated noise by learning a synthetic dataset

- We will be using sparse vector technique!
- For an online sequence of queries
  - Continue to run ”AboveThreshold”, if error is below a noise threshold
    - Return what the synthetic data set returns
  - else: Release the query using Laplace Mechanism
    - Update the synthetic data
    - Restart “AboveThreshold”

Detour: No-regret online learning from expert advice

- N experts, each give advices on stock choices
- After each day, their losses are revealed
- Can I come up with a strategy that does as well as the best expert with (asymptotically) no regret?
- Define **"Regret":**

Multiplicative Weights Algorithm (i.e., the Hedge algorithm)

Algorithm 1 Hedge

1: Initialize: $\forall i \in [N],\ W_1(i) = 1$
2: **for** $t = 1$ to T **do**
3: Pick $i_t \sim_R W_t$, i.e., $i_t = i$ with probability $\mathbf{x}_t(i) = \frac{W_t(i)}{\sum_j W_t(j)}$
4: Incur loss $\ell_t(i_t)$.
5: Update weights $W_{t+1}(i) = W_t(i)e^{-\varepsilon \ell_t(i)}$
6: **end for**

Theorem 1.5. Let ℓ_t^2 denote the N-dimensional vector of square losses, i.e., $\ell_t^2(i) = \ell_t(i)^2$, let $\varepsilon > 0$, and assume all losses to be non-negative. The Hedge algorithm satisfies for any expert $i^\star \in [N]$:

$$\sum_{t=1}^{T} \mathbf{x}_t^\top \ell_t \leq \sum_{t=1}^{T} \ell_t(i^\star) + \varepsilon \sum_{t=1}^{T} \mathbf{x}_t^\top \ell_t^2 + \frac{\log N}{\varepsilon}$$

(From the Online Convex Optimization book by Elad Hazan)

Corollary: we can also compete with the best probability distribution!

Theorem 1.5. Let ℓ_t^2 denote the N-dimensional vector of square losses, i.e., $\ell_t^2(i) = \ell_t(i)^2$, let $\varepsilon > 0$, and assume all losses to be non-negative. The Hedge algorithm satisfies for any expert $i^\star \in [N]$:

$$\sum_{t=1}^{T} \mathbf{x}_t^\top \ell_t \leq \sum_{t=1}^{T} \ell_t(i^\star) + \varepsilon \sum_{t=1}^{T} \mathbf{x}_t^\top \ell_t^2 + \frac{\log N}{\varepsilon}$$

- Why?

How does MW applies to the problem of linear query release?

Online query release without privacy

1. True data $p = x/n$, initial synthetic data $\tilde{p}_1 = 1/|\mathcal{X}|$
2. Adversary selects an online sequence of queries
   - If $|q^T \tilde{p}_t - q^T p| \geq \alpha$
     1. Output $q^T p$
     2. Set the loss vector to be $\ell_t := \mathrm{sign}(q^T \tilde{p}_t - q^T p) \cdot q$
     3. Update $\tilde{p}_{t+1} = \mathrm{Normalize}\big(\tilde{p}_t \cdot \exp(-\eta \ell_t)\big)$
     4. Increment t, i.e., t = t + 1
   - Else: output $q^T \tilde{p}_t$

The regret bound of MW implies that the number of iterations of the MW algorithm is small!

Theorem 1.5. Let ℓ_t^2 denote the N-dimensional vector of square losses, i.e., $\ell_t^2(i) = \ell_t(i)^2$, let $\varepsilon > 0$, and assume all losses to be non-negative. The Hedge algorithm satisfies for any expert $i^\star \in [N]$:

$$\sum_{t=1}^{T} \mathbf{x}_t^\top \ell_t \leq \sum_{t=1}^{T} \ell_t(i^\star) + \varepsilon \sum_{t=1}^{T} \mathbf{x}_t^\top \ell_t^2 + \frac{\log N}{\varepsilon}$$

Private MW for online query release using NumericSparse

Online query release with differential privacy

1. True data $p = x/n$, initial synthetic data $\tilde{p}_1 = 1/|\mathcal{X}|$
2. Adversary selects an online sequence of queries
   - If $|q^T \tilde{p}_t - q^T p| \geq \alpha$ ← Use AboveThresh for this
     1. Output $q^T p$ ← Use Laplace mechanism
     2. Set the loss vector to be $\ell_t := \text{sign}(q^T \tilde{p}_t - q^T p) \cdot q$
     3. Update $\tilde{p}_{t+1} = \text{Normalize}\big(\tilde{p}_t \cdot \exp(-\eta \ell_t)\big)$
     4. Increment t, i.e., t = t + 1
   - Else: output $q^T \tilde{p}_t$

Private MW for online query release using NumericSparse

Online query release with differential privacy

1. True data $p = x/n$, initial synthetic data $\tilde{p}_1 = 1/|\mathcal{X}|$
2. Adversary selects an online sequence of queries
   $\hat{\alpha} = \alpha + \mathrm{Lap}(2/(n\epsilon_0))$
   - If $|q^T \tilde{p}_t - q^T p| + \mathrm{Lap}(4/(n\epsilon_0)) \geq \hat{\alpha}$
     1. Privately release $y = q^T p + \mathrm{Lap}(1/(n\epsilon_0))$
     2. Set the loss vector to be $\ell_t := \mathrm{sign}(q^T \tilde{p}_t - y) \cdot q$
     3. Update $\tilde{p}_{t+1} = \mathrm{Normalize}\big(\tilde{p}_t \cdot \exp(-\eta \ell_t)\big)$
     4. Increment t, i.e., t = t + 1. Break if t > N
     5. Refresh threshold noise: $\hat{\alpha} = \alpha + \mathrm{Lap}(2/(n\epsilon_0))$
   - Else: output $q^T \tilde{p}_t$

Privacy analysis is straightforward.

- The algorithm runs AboveThresh + Laplace Mechanism for at most N times.
  - Total privacy loss bounded by $2N\varepsilon_0$
  - We could choose $2N\varepsilon_0 = \varepsilon_{\text{budget}}$

- Note unique. How to choose N,ε_0?
  - We need to choose ε_0 s.t. the accuracy criteria is met.
  - We need to guess (and bound) the number of iterations the Hedge algorithms will need to run.
  - Choose one pair of N,ε_0 that works.

Utility analysis of the private MW Mechanism

1. Bound all Laplace random variables (how many are they?)

2. All that are not selected are getting accurate answers

3. All that are selected are also getting accurate answers

4. From the regret bound of MW, the number of iterations is small

Summarize the result into a theorem statement

- Choose these parameters
  - ε_0 =
  - N =

Theorem (Utility of Private MW): With probability at least 1- δ, The private MW algorithm calibrated to achieve with ε-DP is able to answer any online sequence of |Q| linear queries and a max error of:

Remainder of today's lecture

- Introducing the problem of private selection
- Exponential mechanism
- The privacy analysis of exponential mechanism

Private selection

- A (large) set of items, and a utility function.

$$\mathcal{R} \qquad\qquad u : \mathbb{N}^{\mathcal{X}} \times \mathcal{R} \to \mathbb{R}$$

- Example 1 (Most popular movie)
- Example 2 (Learning a classifier)
- Example 3 (Auction)

Exponential mechanism

- Global sensitivity of the utility function

$$\Delta u \equiv \max_{r \in \mathcal{R}} \max_{x,y:\|x-y\|_1 \leq 1} |u(x,r) - u(y,r)|.$$

- The exponential mechanism samples an output from a "Gibbs distribution":

$$\mathcal{M}(x) \sim p(r|x) \propto \exp\left(\frac{\varepsilon u(x,r)}{2\Delta u}\right)$$

Privacy Analysis of Exponential Mechanism

Randomized response and Laplace mechanism are instances of exponential mechanisms!

Next lecture

- Utility analysis of exponential mechanism
- Report Noisy Max
- Privacy loss random variable and advanced composition