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

Yu-Xiang Wang
Recap: last lecture

• Generalizing the problem of linear query release
  \[ \frac{1}{n} \sum_{i=1}^{n} f(x_i) = \frac{1}{n} \langle q_f, x \rangle \]

• Apply Laplace mechanism to this problem
  • Releasing queries
    \[ \epsilon_{\text{query}} \leq \left( \frac{1}{n} \right) \frac{k \log \frac{k}{\delta}}{\epsilon} \quad \text{w.p. 1 - } \delta \]
  • Releasing data (i.e., contingency table)
    \[ x \overset{\Delta}{=} x + \text{lap}(\frac{1}{\epsilon}) \quad \left( \frac{1}{n} \| G^T x - q_f^T x \| \right) \leq \frac{\sqrt{1/k} \log \frac{k}{\delta}}{\epsilon} \quad \text{w.p. 1 - } \delta \]

• 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 $a_1, \ldots$

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

Let $\hat{T} = T + \text{Lap}\left(\frac{2}{\epsilon}\right)$.

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

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

Any noise-adding mechanism $M_2$ satisfying $\epsilon/2$-DP for queries with sensitivity 2.
Recap: SparseVector mechanism

1. Start with a budget of $\varepsilon$, and a maximum number of “discoveries” $c$.
2. Split $\varepsilon$ into $c$ equal parts and run \textit{AboveThreshold} for up to $c$ times, each with a privacy budget of $\varepsilon/c$.
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
   \[ \{ \bot^k \top | k = 0, 1, \ldots, \infty \} \]
   - 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
   \[
   \Pr[M(D) = k] = \mathbb{E}_{z \sim p_\rho} [\Pr[M(D) = k | z]]
   = \mathbb{E}_{z \sim p_\rho} [\prod_{i \leq k} \Pr[q_i(D) + \nu_i < z | z] \Pr[q_{k+1}(D) + \nu_i \geq z | z]]
   = \int_{-\infty}^{+\infty} \rho(z) \left( \prod_{i \leq k} \int_{-\infty}^{z-q_i(D)} p(\nu_i) d\nu_i \right) \cdot \int_{z-q_{k+1}(D)}^{\infty} p(\nu_{k+1}) d\nu_{k+1} dz
   \]
Recap: The analysis of AboveThreshold

\[
\Pr[\mathcal{M}(D) = k] = \mathbb{E}_{z \sim p_\rho} \left[ \Pr[\mathcal{M}(D) = k | z] \right] 
= \mathbb{E}_{z \sim p_\rho} \left[ \prod_{i \leq k} \Pr[q_i(D) + \nu_i < z | z] \Pr[q_{k+1}(D) + \nu_i \geq z | z] \right] 
= \int_{-\infty}^{+\infty} p_\rho(z) \left( \prod_{i \leq k} \int_{-\infty}^{z-q_i(D)} p(\nu_i) d\nu_i \right) \cdot \int_{z-q_k+1(D)}^{\infty} p(\nu_{k+1}) d\nu_{k+1} 
\]

Key trick: a change of variable that shifts noisy-threshold by \( \Delta \)

\[
u := z + \Delta + \infty \quad \downarrow \quad \int_{-\infty}^{+\infty} p_\rho(\nu - \Delta) \left( \prod_{i \leq k} \int_{-\infty}^{\nu - \Delta - q_i(D)} p(\nu_i) d\nu_i \right) \cdot \int_{\nu - \Delta - q_{k+1}(D)}^{\infty} p(\nu_{k+1}) d\nu_{k+1} d\nu 
= \int_{-\infty}^{+\infty} p_\rho(u) \left( \frac{p_\rho(u - \Delta)}{p_\rho(u)} \right) \left( \prod_{i \leq k} \int_{-\infty}^{u - \Delta - q_i(D)} p(\nu_i) d\nu_i \right) \cdot \int_{u - \Delta - q_{k+1}(D)}^{\infty} p(\nu_{k+1}) d\nu_{k+1} 
= \mathbb{E}_{z \sim p_\rho} \left[ \left( \frac{p_\rho(z - \Delta)}{p_\rho(z)} \right) \left( \prod_{i \leq k} \int_{-\infty}^{z - \Delta - q_i(D)} p(\nu_i) d\nu_i \right) \cdot \int_{z - \Delta - q_{k+1}(D)}^{\infty} p(\nu_{k+1}) d\nu_{k+1} \right] 
\]

\[
P_r(\mathcal{Z} = z - \Delta) \leq \Pr(\Delta + \mathcal{Z} = 8) \quad P(\nu_i \leq z - \Delta \theta q_i(D)) \\
P_r(\mathcal{Z} = z) = P(0 \in \mathcal{Z} = 8) \\
P_r(\nu_i \leq z + \Delta \theta q_i(D)) \\
P(\nu_i \theta q_i(D) \triangleq z \leq 8) \leq \Pr(\nu_i \theta q_i(D) \leq 8)
\]
Recap: Bounding the third term via a fictitious query

\[ \int_{z-\Delta-q_{k+1}(D)}^{\infty} p(\nu_{k+1}) d\nu_{k+1} \]

\[ \Pr [V_{k+1} \geq \tilde{q}_k(D) + \Delta] \leq e^\epsilon \Pr [q_{k+1}(D') + V_{k+1} \geq 2] \]

• 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 \( \tilde{q} \)? \( \geq \Delta \)

\[ \forall k \]

\[ \forall c \]

\[ \Pr [M(D) = c] \leq e^\epsilon \Pr [M(D') = c] \]
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
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.
  \[ q_i^T x + v_i \geq T + \varepsilon \quad \text{for all } i \text{ such that } 0 < \frac{v_i}{x_i} \leq \frac{2c}{\varepsilon} \]
  - (No-false discovery). With high probability, we also do not wrongly identify queries that are not interesting as interesting.
  \[ q_i^T x < T - \frac{8c \log 2k}{\varepsilon} \]
- 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 $O(polylog (|Q|, |X|))$ 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”:

\[
\sum_{t=1}^{T} \langle x_t, l_t \rangle - \min_{i} \sum_{t=1}^{T} l_t[i] = o(T)
\]

where \(x_t \in \Delta^n\), \(l_t \in [0, 1]^n\).
Multiplicative Weights Algorithm (i.e., the Hedge algorithm)

Algorithm 1 Hedge

1: Initialize: \( \forall i \in [N], \ W_1(i) = 1 \)
2: \( \textbf{for} \ t = 1 \ \textbf{to} \ T \ \textbf{do} \)
3: \( \text{Pick } i_t \sim_R W_t, \ i.e., \ i_t = i \ \text{with probability } \ x_t(i) = \frac{W_t(i)}{\sum_j W_t(j)} \)
4: \( \text{Incur loss } l_t(i_t), \ l_t(x_t) = l_t^T x_t \)
5: \( \text{Update weights } W_{t+1}(i) = W_t(i) e^{-c_l(i)} \)
6: \( \text{end for} \)

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

\[
\sum_{t=1}^T x_t^T l_t \leq \sum_{t=1}^T l_t(i^*) + \varepsilon \sum_{t=1}^T x_t^T l_t^2 + \frac{\log N}{\varepsilon}
\]

(From the Online Convex Optimization book by Elad Hazan)
Corollary: we can also compete with the best probability distribution!

Theorem 1.5. Let $\ell_t^2$ 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 $i^* \in [N]$:

$$\sum_{t=1}^{T} x_t^\top \ell_t \leq \sum_{t=1}^{T} \ell_t(i^*) + \varepsilon \sum_{t=1}^{T} x_t^\top \ell_t^2 + \frac{\log N}{\varepsilon}$$

• Why?

$$\sum_{t=1}^{T} x_t^\top \ell_t \leq \sum_{i=1}^{T} p_t^i \ell_t + 2\sqrt{T \log N}$$
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 \( \ell_t := \text{sign}(q^T \tilde{p}_t - q^T p) \cdot q \)
     3. Update \( \tilde{p}_{t+1} = \text{Normalize}(\tilde{p}_t \cdot \exp(-\eta \ell_t)) \)
     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 $\ell_t^2$ 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 $i^* \in [N]$: 

$$
\sum_{t=1}^{T} x_t^T \ell_t \leq \sum_{t=1}^{T} \ell_t(i^*) + \varepsilon \sum_{t=1}^{T} x_t^T \ell_t^2 + \frac{\log N}{\varepsilon}
$$
Private MW for online query release using **NumericSparse**

**Online query release with differential privacy**

1. True data \( p = x/n \), initial synthetic data \( \tilde{p}_1 = 1/|X| \)
2. Adversary selects an online sequence of queries
   - If \( |q^T \tilde{p}_t - q^T p| \geq \alpha \) \( \Rightarrow \) Use AboveThresh for this
     1. Output \( q^T p \) \( \Rightarrow \) 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 \( \tilde{p}_{t+1} = \text{Normalize}(\tilde{p}_t \cdot \exp(-\eta \ell_t)) \)
     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 \( \tilde{p}_1 = 1/|X| \)
2. Adversary selects an online sequence of queries
   \[
   \hat{\alpha} = \alpha + \text{Lap}\left(\frac{2}{(n\epsilon_0)}\right)
   \]
   - If \(|q^T \tilde{p}_t - q^T p| + \text{Lap}\left(\frac{4}{(n\epsilon_0)}\right) \geq \hat{\alpha}\)
     1. Privately release \( y = q^T p + \text{Lap}\left(\frac{1}{(n\epsilon_0)}\right) \)
     2. Set the loss vector to be \( \ell_t := \text{sign}(q^T \tilde{p}_t - y) \cdot q \)
     3. Update \( \tilde{p}_{t+1} = \text{Normalize}(\tilde{p}_t \cdot \exp(-\eta \ell_t)) \)
     4. Increment \( t \), i.e., \( t = t + 1 \). Break if \( t > N \)
     5. Refresh threshold noise: \( \hat{\alpha} = \alpha + \text{Lap}\left(\frac{2}{(n\epsilon_0)}\right) \)
   - 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 $N, \varepsilon_0$?
  - We need to choose $\varepsilon_0$ 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 $N, \varepsilon_0$ that works.
Utility analysis of the private MW Mechanism

1. Bound all Laplace random variables (how many are they?)
   \[ 2N + k \text{ lap RV. } \frac{4}{n \varepsilon_0} \quad \text{w.p. } 1 - \delta, \text{ all of them } |Z| \leq \frac{4}{n \varepsilon_0} \log \frac{2N + k}{\delta} \]

2. All that are not selected are getting accurate answers
   Rule: \[ q_i^{\ast} p_t - q_i p_t \leq \alpha + \delta \quad \Rightarrow \quad |q_i^{\ast} p_t - q_i p_t| \leq \alpha + \delta + \frac{1}{2} \]

3. All that are selected are also getting accurate answers
   Rule: \[ |q_i^{\ast} p_t - q_i p_t| - e_i \geq \alpha + \delta \Rightarrow \text{"T"} \Rightarrow q_i^{\ast} p_t + w \Rightarrow |u| \leq \frac{4}{n \varepsilon_0} \left( \frac{8}{3k} \right) \]

4. From the regret bound of MW, the number of iterations is small
   \[ \text{Regret} = \sum_i y_i p_t^{\ast} - p_t^{\ast} \sum_i (q_t^{\ast} - q_t) \cdot \text{Sign} \left( \frac{y_t}{y_t^{\ast}} + \beta q_t + w \right) \leq \frac{M^2 \varepsilon_0}{2} - p_t^{\ast} \]
   \[ \text{Condition on } \text{"T"}, \quad |q_i^{\ast} p_t - q_i p_t| \geq \alpha + \delta - e_i \geq \frac{\alpha}{2} \]
   \[ -\frac{\alpha}{2} \leq 2T \frac{y_t}{y_t^{\ast}} \leq \frac{16 (\alpha + \delta)}{\alpha^2} \]
   \[ T \leq \frac{1}{\alpha^2} \]
Summarize the result into a theorem statement

- Choose these parameters
  - $\varepsilon_0 = \frac{16 \log \frac{3k}{\delta}}{n \alpha}
  - $N = \frac{16 \log |X|}{\alpha^2}$
  - $\varepsilon_{total} = 2 \varepsilon_0 N = \frac{512 \log \frac{3k}{\delta} \log |X|}{n \alpha^3}$

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

$$\alpha \leq 1.25 \left( \frac{512 \log \frac{3k}{\delta} \log |X|}{n \varepsilon} \right)$$
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.
  \[ \mathcal{R} \quad u : \mathbb{N}^x \times \mathcal{R} \rightarrow \mathbb{R} \]

• Example 1 (Most popular movie)
  \[ u(x, r) = \text{argmax}_{r \in \mathcal{R}} u(x, r) \]

• Example 2 (Learning a classifier)
  \[ R := \{ \text{Decision-free } \theta, \text{ Linear classifier } \theta, \text{ LC } \theta, \ldots \} \]
  \[ u(x, r) = -\mathbb{E}_{\text{new}(r)} \mathbb{E}_{\text{train}(r)} \]

• Example 3 (Auction)
  \[ u(x, r) = r \cdot \text{# of people who愿意 buy} \]
Exponential mechanism

• Global sensitivity of the utility function

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

• 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

\[
\frac{P_x(r)}{P_x(r')} = e^{\frac{\epsilon u(x, r)}{\Delta}} \geq e^{\frac{\epsilon u(x, r')}{\Delta}} = e^{\frac{\epsilon (u(x, r) - u(x, r'))}{\Delta}} \geq e^{\frac{\epsilon u(x, r')}{\Delta}} \geq e^{\frac{\epsilon u(x, r')}{\Delta}}
\]
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