Advanced Topics in Machine Learning 2024

Amartya Sanyal

Home Assignment 3

Deadline: 18:00, Thursday, 3 October 2024

The assignments must be submitted individually – each student must write and submit a personal solution but we do not prevent you from discussing ***high-level*** *ideas in small groups. If you use any LLM tool such as ChatGPT, please specify the* ***purpose*** *and* ***manner*** *in which you have utilized it.*

We are interested in how *you solved the problems, and not just in the final answers. Please explain your solutions and ideas as clearly as you can.*

Late Penalty and multiple Submissions *Late submissions will incur a penalty of 10% of the total marks for every hour of delay (rounded up) with a maximum allowed delay of 5 hours after which the submission server will close. If you submit multiple submissions, only the last submission will be considered relevant both for grading answers as well as late penalty.*

Submission format: *Please upload your answers in a single* `.pdf` *file. If you have created code please include at least the key parts in the PDF as well as a link to the full code (on Github, Colab, or similar).*

Lets consider the following voting system present in a mythical country called Denland. There are n electoral candidates and m voters. Each voter votes for as many of the n candidates as they choose. Then, the total count of votes for candidate i is calculated and represented as c_i. We refer to this list of counts as $\mathcal{C} \in \mathbb{N}^n$. In the winner-takes-all system, the candidate with the maximum votes *i.e.* $i^* = \arg\max_i c_i$ wins. Of course, we want to do this privately and the government of Denland is looking for ways to privately release the counts of candidates.

Question 1. *We begin this journey by first designing an algorithm to private release the amount of votes each candidate gathered.*

1. ***5 points*** *Design an algorithm that inputs the voting preferences of all candidates and releases a private list of count of votes, denoted as $\mathcal{C}_\epsilon \in \mathbb{N}^n$, where each entry in the list corresponds to the number of votes for each candidate. The output list should satisfy ϵ-DP guarantee with respect to the preference list of each voter.*

2. ***5 points*** *Prove the privacy of the algorithm.*

3. ***10 points*** *Compute the expected (expectation is over the randomness in the algorithm) L_1 norm between the original counts and the released counts i.e. $\mathbb{E}\left[\|\mathcal{C}_\epsilon - \mathcal{C}\|_1\right]$*

The Government of Denland is not happy with the above guarantees and wants to improve it. A group of students in the University of Denland was assigned the task of coming up with a better algorithm and they proposed the following algorithm (see Alg 1).

Question 2. *However, they did not provide guarantees for the algorithm, which we are now supposed to do in this exercise problem.*

1. ***30 points*** *State and prove the Differential Privacy guarantee of Algorithm 1.*
   Hint: *The algorithm samples a n dimensional noisy vector. Fix the elements of that vector for all but the argmax location for both neighbouring datasets and then prove the DP equalities conditioned on that. Then remove the conditioning using standard probability tools.*

Algorithm 1 Private voting in Denland

```
Input: Total votes $c_1, c_2, \ldots, c_n$, Privacy parameter $\epsilon$
for each $i = 1, \ldots, n$ do
    Sample noise $Z_i \sim \text{Laplace}\left(\frac{2}{\epsilon}\right)$
    Compute noisy total votes: $\tilde{c}_i = c_i + Z_i$
end for
Select index $i^* = \arg\max_i \tilde{q}_i$
Return $i^*$
```

2. ***30 points*** *Let $i^\star$ be the output of Algorithm 1 and $j^\star$ be the true winner of the election. Prove the following utility guarantee*

$$\mathbb{P}\left[c_{i^\star} < c_{j^\star} - \frac{4(\log n + t)}{\epsilon}\right] \leq \exp(-t).$$

For this exercise, you will first have to prove the following two inequalities for a Laplacian random vector $Y \sim \text{Lap}(\lambda)$:

   (a) For any $t \geq 0$, prove $\mathbb{P}(|Y| \geq \lambda t) \leq \exp(-t)$.

   (b) Let $Y_1, \ldots, Y_k \sim \text{Lap}(\lambda)^k$. Define $Y_{\max} = \max_{i \leq k} Y_i$. Then, using the above prove that $\mathbb{P}(|Y_{\max}| > \lambda(\log(k) + t)) \leq \exp(-t)$

Now use these results to prove the utility guarantee.

Question 3. *Now its time to reflect on these algorithms and see why one would be more preferable than the other.*

1. ***5 points*** *Compare the algorithms in Question 2 and 1 and explain why such seemingly similar algorithms shows different performance.*
2. ***10 points*** *Show how to apply Exponential mechanism from class to the problem in Question 2. Then prove that Exponential mechanism achieves the same utility guarantee as the bound proved in Question 2 part 2..*
3. ***5 points*** *Comment on when you would prefer one over the other (all three algorithms).*

Good luck!
Amartya