Catalog

Question 1 1
1. Answer: 1
2. Answer: 2
3. Answer: 3
Question 2 3
1. Answer: 3
2. Answer: 5
Proof of (a): 5
Proof of (b): 5
Proof of the utility guarantee: 5
Question 3 7
1. Answer: 7
2. Answer: 8
3. Answer: 9

Question 1

1. Answer:

Algorithm Design

Input: The voting preferences of all m voters, where each voter votes for any subset of the n candidates.

Output: A private list of vote counts $C_\varepsilon \in \mathbb{N}^n$, where each entry corresponds to the number of votes for each candidate, with differential privacy guarantee ε.

Steps:

1: Compute True Counts:

For each candidate $i \in \{1,2,\ldots,n\}$, calculate the true vote count c_i by summing the number of voters who voted for candidate i.

2: Add Laplace Noise:

Determine the global sensitivity Δf of the counting function. Since each voter can affect the counts of up to n candidates by at most 1, $\Delta f = n$.

For each candidate i, generate Laplace noise $\eta_i \sim$ Laplace $\left(\frac{n}{\varepsilon}\right)$.

Compute the noisy count:$c_i^\varepsilon = c_i + \eta_i$

Optionally, enforce non-negativity by setting $c_i^\varepsilon = \max\{0, c_i^\varepsilon\}$.

3: Output Noisy Counts:

Return the vector $C_\varepsilon = (c_1^\varepsilon, c_2^\varepsilon, \dots, c_n^\varepsilon)$.

2. Answer:

Privacy Proof

Prove that the algorithm satisfies ε-differential privacy with respect to the preference list of each voter.

Definition of differential privacy:

An algorithm A is ε-differentially private if for all pairs of neighboring datasets D and D' (differing in the data of one individual voter), and for all measurable subsets S of the output space:

$$\Pr[A(D) \in S] \leq e^\varepsilon \Pr[A(D') \in S]$$

Global sensitivity calculation:

Function: The function f maps the dataset D to the counts $C \in \mathbb{N}^n$.

Neighboring datasets: Two datasets D and D' are neighboring if they differ in the votes of a single voter.

Sensitivity Δf: The maximum change in the counts when one voter's data changes. Since a voter can change their vote for each of the n candidates (from not voting to voting or vice versa), the maximum change in the counts is n (in L_1 norm).

Applying the Laplace Mechanism:

The Laplace mechanism adds noise drawn from the Laplace distribution with scale parameter $b = \frac{\Delta f}{\varepsilon} = \frac{n}{\varepsilon}$.

Adding independent Laplace noise with this scale to each count ensures ε-differential privacy.

Proof:

For any two neighboring datasets D and D', and any output vector C_ε:$\frac{\Pr[A(D)=C_\varepsilon]}{\Pr[A(D')=C_\varepsilon]} \leq e^\varepsilon$

This inequality holds because the Laplace mechanism's probability density functions for D and D' differ by at most a factor of e^ε due to the sensitivity $\Delta f = n$ and the noise scale

$b = \frac{n}{\varepsilon}$.

3. Answer:

Compute the expected value of the L_1 norm between the original counts C and the released counts C_ε:

$$\mathbb{E}[\|C_\varepsilon - C\|_1] = \sum_{i=1}^{n} \mathbb{E}[|c_i^\varepsilon - c_i|]$$

Since $c_i^\varepsilon = c_i + \eta_i$, where η_i is Laplace noise with scale $b = \frac{n}{\varepsilon}$, the difference $|c_i^\varepsilon - c_i|$ is $|\eta_i|$.

The expected absolute value of a Laplace random variable with scale b is: $\mathbb{E}[|\eta_i|] = b = \frac{n}{\varepsilon}$

Therefore, the expected L_1 norm is:$\mathbb{E}[\|C_\varepsilon - C\|_1] = \sum_{i=1}^{n} \mathbb{E}[|\eta_i|] = n \times \frac{n}{\varepsilon} = \frac{n^2}{\varepsilon}$

Question 2

1. Answer:

Differential Privacy Guarantee of Algorithm 1

Assumptions:

Voting Behavior: Each voter votes for at most one candidate. This assumption is crucial because it limits the maximum change a single voter can make to the vote counts.

Global Sensitivity $\Delta f = 2$: A single voter's data can change the counts of at most two candidates by ± 1 (voting for one candidate and not voting for another).

Noise scale: Laplace noise with scale $\lambda = \frac{2}{\varepsilon}$ is added to each count.

Privacy proof:

Show that Algorithm 1 satisfies ε-differential privacy, i.e., for any two neighboring datasets D and D' differing in one voter's data, and for any output i:

$$\Pr[i^*(D) = i] \leq e^\varepsilon \Pr[i^*(D') = i]$$

1.Neighboring datasets

Let D and D' be neighboring datasets differing in one voter's vote.

The counts $c = (c_1, \dots, c_n)$ and $c' = (c'_1, \dots, c'_n)$ differ in at most two positions, say candidates i and j:

$c_i = c'_i + 1$,

$c_j = c'_j - 1$,

For all other $k \neq i, j$, $c_k = c'_k$.

2.Fix noise for other candidates

Fix the noise Z_k for all $k \neq i, j$.

Let $M = \max_{k \neq i,j}(c_k + Z_k)$. This value is the same for both D and D'.

3.Compute conditional probabilities

For dataset D:

$\tilde{c}_i = c_i + Z_i$

$\tilde{c}_j = c_j + Z_j$

For dataset D':

$\tilde{c}'_i = c'_i + Z_i = (c_i - 1) + Z_i = \tilde{c}_i - 1$

$\tilde{c}'_j = c'_j + Z_j = (c_j + 1) + Z_j = \tilde{c}_j + 1$

Probability of output i:

$\Pr[i^* = i | Z_{-i,j}] = \Pr[\tilde{c}_i \geq \max(M, \tilde{c}_j)]$

$\Pr[i^{*'} = i | Z_{-i,j}] = \Pr[\tilde{c}_i - 1 \geq \max(M, \tilde{c}_j + 1)]$

4.Ratio of probabilities

Compute the ratio of the probabilities conditioned on $Z_{-i,j}$:

$$\frac{\Pr[i^* = i | Z_{-i,j}]}{\Pr[i^{*'} = i | Z_{-i,j}]} = \frac{\Pr[\tilde{c}_i \geq T]}{\Pr[\tilde{c}_i \geq T + 2]}$$

where $T = \max(M, \tilde{c}_j)$.

5.Bounding the ratio using Laplace distribution

Since $\tilde{c}_i = c_i + Z_i$, the noise Z_i follows Laplace$(2/\varepsilon)$.

The tail of the Laplace distribution satisfies:

For any $t \geq 0$:$\Pr[Z_i \geq t] = \frac{1}{2}\exp\left(-\frac{\varepsilon t}{2}\right)$

Therefore, the ratio becomes:$\frac{\Pr[Z_i \geq s]}{\Pr[Z_i \geq s+2]} = \exp\left(\frac{\varepsilon \cdot 2}{2}\right) = e^{\varepsilon}$

Thus:$\frac{\Pr[i^*=i|Z_{-i,j}]}{\Pr[i^{*'}=i|Z_{-i,j}]} \leq e^{\varepsilon}$

6.Unconditional probabilities

Since the inequality holds for all fixed $Z_{-i,j}$, it also holds unconditionally after integrating over $Z_{-i,j}$:$\Pr[i^* = i] \leq e^{\varepsilon}\Pr[i^{*'} = i]$

Algorithm 1 satisfies ε-differential privacy under the assumption that each voter votes for at most one candidate.

2. Answer:

Proof of (a):

For $Y \sim$ Laplace(λ) and any $t \geq 0$: $\Pr[|Y| \geq \lambda t] \leq e^{-t}$

The Laplace distribution has the probability density function:$f_Y(y) = \frac{1}{2\lambda}\exp\left(-\frac{|y|}{\lambda}\right)$

The cumulative distribution function for $|Y| \geq \lambda t$ is:$\Pr[|Y| \geq \lambda t] = 2\left(\frac{1}{2}\exp\left(-\frac{\lambda t}{\lambda}\right)\right) = e^{-t} = exp(-t)$

Proof of (b):

For $Y_1, Y_2, \dots, Y_k \sim$ Laplace(λ) and $Y_{\max} = \max_i |Y_i|$:

$$P(Y_{\max} \geq \lambda(\log\, k + t)) \leq e^{-t}$$

Using the union bound:$P(Y_{\max} \geq \lambda(\log\, k + t)) \leq \sum_{i=1}^{k} P(|Y_i| \geq \lambda(\log\, k + t))$

From part (a):$P(|Y_i| \geq \lambda(\log\, k + t)) = e^{-(\log\, k+t)} = \frac{1}{k}e^{-t}$

Summing over k variables:$P(Y_{\max} \geq \lambda(\log\, k + t)) \leq k \cdot \frac{1}{k}e^{-t} = e^{-t} = exp(-t)$

Proof of the utility guarantee:

1.Define:

Let $Z_i \sim$ Laplace$\left(\frac{2}{\varepsilon}\right)$.

Define $\Delta = \frac{4(\ln\, n+t)}{\varepsilon}$.

To show:$\Pr\left(c_{i^*} < c_{j^*} - \Delta\right) \leq e^{-t}$

2.Analyze the noisy counts:

The algorithm selects $i^* = \arg\ \max_i (c_i + Z_i)$.

The true winner is $j^* = \arg\ \max_i c_i$.

3.Bound the noise differences:

For $i \neq j^*$, $c_{j^*} - c_i \geq 0$.

The difference in noisy counts:$(c_{j^*} + Z_{j^*}) - (c_i + Z_i) = (c_{j^*} - c_i) + (Z_{j^*} - Z_i)$

Want $(c_{j^*} - c_i) + (Z_{j^*} - Z_i) > 0$ to ensure $i^* = j^*$.

4.Probability of incorrect winner:

The probability that $i^* \neq j^*$ is:$\Pr\left[\exists i \neq j^* \text{ such that } (c_i + Z_i) \geq (c_{j^*} + Z_{j^*})\right]$

This is equivalent to:$\Pr\left[\exists i \neq j^* \text{ such that } (c_{j^*} - c_i) \leq Z_i - Z_{j^*}\right]$

5.Bounding the noise differences:

For each $i \neq j^*$, $Z_i - Z_{j^*}$ is a random variable with probability density function (since Z_i and Z_{j^*} are independent and identically distributed):$\Pr[Z_i - Z_{j^*} \geq s] = \Pr[\text{Laplace}\left(0, \frac{4}{\varepsilon}\right) \geq s]$

Using the tail bound from part (a), for $s \geq 0$:$\Pr[Z_i - Z_{j^*} \geq s] \leq \exp\left(-\frac{\varepsilon s}{4}\right)$

6.Applying union bound:

The probability that any $i \neq j^*$ satisfies $(c_{j^*} - c_i) \leq Z_i - Z_{j^*}$ is bounded by:$\sum_{i \neq j^*} \Pr[Z_i - Z_{j^*} \geq -(c_{j^*} - c_i)]$

Since $c_{j^*} - c_i \geq 0$, we have:$\Pr[Z_i - Z_{j^*} \geq -(c_{j^*} - c_i)] \leq \Pr[|Z_i - Z_{j^*}| \geq c_{j^*} - c_i]$

Using part (b) with $k = n - 1$, the maximum of $n - 1$ such terms is:$\Pr\left[\max_{i \neq j^*} |Z_i - Z_{j^*}| \geq \frac{4(\log(n-1) + t)}{\varepsilon}\right] \leq e^{-t}$

Conclusion

Therefore:$\Pr\left[c_{j^*} - c_{i^*} \geq \frac{4(\log n + t)}{\varepsilon}\right] \geq 1 - e^{-t}$ which implies: $\Pr\left[c_{i^*} < c_{j^*} - \frac{4(\log n + t)}{\varepsilon}\right] \leq e^{-t}$

That is: $Pr[c_{i^*} < c_{j^*} - \frac{4(\log n + t)}{\epsilon}] \leq \exp(-t)$

Question 3

1. Answer:

Comparison of the algorithms in questions 1 and 2

Overview of the algorithms:

Algorithm from question 1:

Scenario: Each voter can vote for as many candidates as they choose.

Goal: Privately release the count of votes for each candidate.

Method:

Compute true vote counts c_i for each candidate.

Add Laplace noise with scale $\lambda = \frac{n}{\varepsilon}$ to each count.

Output the noisy counts $c_i^{\varepsilon} = c_i + \eta_i$.

Privacy Guarantee: Satisfies ε-differential privacy with global sensitivity $\Delta f = n$.

Expected Error: $\mathbb{E}[\|C_{\varepsilon} - C\|_1] = \frac{n^2}{\varepsilon}$.

Algorithm from Question 2:

Scenario: Each voter votes for at most one candidate.

Goal: Privately select the winning candidate.

Method:

Compute true vote counts c_i.

Add Laplace noise with scale $\lambda = \frac{2}{\varepsilon}$ to each count.

Select $i^* = \arg\,\max_i (c_i + Z_i)$.

Privacy guarantee: satisfies ε-differential privacy with global sensitivity $\Delta f = 2$.

Utility guarantee: The probability that the selected winner's true vote count is significantly less than the true winner's count decreases exponentially with t:$\Pr\left[c_{i^*} < c_{j^*} - \frac{4(\ln\, n+t)}{\varepsilon}\right] \le e^{-t}$

Reasons for Different Performance:

1.Global sensitivity differences:

Algorithm 1: The global sensitivity $\Delta f = n$ because a single voter can affect the counts of all n candidates by ± 1.

Algorithm 2: The global sensitivity $\Delta f = 2$ because a single voter can change their vote from one candidate to another, affecting at most two candidates by ± 1.

2.Noise scale differences:

Algorithm 1: Adds noise with a larger scale $\lambda = \frac{n}{\varepsilon}$, leading to higher overall noise.

Algorithm 2: Adds noise with a smaller scale $\lambda = \frac{2}{\varepsilon}$, resulting in less distortion.

3.Performance implications:

Algorithm 1: Provides noisy counts for all candidates but with higher expected error due to larger noise, especially when n is large.

Algorithm 2: Focuses on selecting the winner with higher accuracy and provides strong utility guarantees for the winning candidate, with less noise added.

Conclusion: The key difference lies in the assumptions about voter behavior and the resulting global sensitivity. Algorithm 2 achieves better performance in winner selection due to lower sensitivity, allowing it to add less noise while maintaining privacy. Despite both algorithms adding Laplace noise to vote counts, the differences in sensitivity and goals (releasing counts vs. selecting a winner) lead to different performances.

2. Answer:

Applying the Exponential Mechanism to the Problem in Question 2

Exponential mechanism overview:

Purpose: Selects an output that balances utility and privacy by sampling from a probability distribution that favors higher utility outputs while satisfying differential privacy.

Mechanism: For a utility function $u(D, r)$, the mechanism selects output r with probability proportional to $\exp\left(\frac{\varepsilon u(D,r)}{2\Delta u}\right)$, where Δu is the sensitivity of the utility function.

Applying to the problem:

Utility function: Define $u(D, i) = c_i$, where c_i is the true vote count for candidate i.

Sensitivity of Utility Function: Changing one voter's vote can change c_i by at most 1, so $\Delta u = 1$.

Selection probability: The probability of selecting candidate i:$P[i] = \frac{\exp\left(\frac{\varepsilon c_i}{2}\right)}{\sum_{j=1}^{n} \exp\left(\frac{\varepsilon c_j}{2}\right)}$

Mechanism steps: 1. Compute $\exp\left(\frac{\varepsilon c_i}{2}\right)$ for each candidate. 2. Normalize the probabilities to sum to 1. 3. Sample a candidate i^* according to the computed probabilities.

Proof of the utility guarantee:

To show that the Exponential Mechanism achieves a utility guarantee similar to Algorithm 1:

$$\Pr\left[c_{i^*} < c_{j^*} - \Delta\right] \leq e^{-t}$$

where $\Delta = \frac{2\log n+2t}{\varepsilon}$.

Proof:

Denominator lower bound: The sum in the denominator:$S = \sum_{j=1}^{n} \exp\left(\frac{\varepsilon c_j}{2}\right) \geq \exp\left(\frac{\varepsilon c_{j^*}}{2}\right)$

Numerator upper bound: The sum over candidates with counts $c_i \leq c_{j^*} - \Delta$:$N = \sum_{c_i \leq c_{j^*}-\Delta} \exp\left(\frac{\varepsilon c_i}{2}\right) \leq n\exp\left(\frac{\varepsilon(c_{j^*}-\Delta)}{2}\right)$

Probability bound: The probability of selecting a candidate with $c_i \leq c_{j^*} - \Delta$:$\Pr\left[c_{i^*} \leq c_{j^*} - \Delta\right] = \frac{N}{S} \leq n\exp\left(-\frac{\varepsilon\Delta}{2}\right)$

Substituting Δ: Let $\Delta = \frac{2\log n+2t}{\varepsilon}$: $\Pr\left[c_{i^*} \leq c_{j^*} - \Delta\right] \leq n\exp\left(-\frac{\varepsilon(2\log n+2t)}{2\varepsilon}\right) = e^{-t}$

Conclusion: The exponential mechanism achieves the same utility guarantee as Algorithm 1, with the probability of selecting a candidate whose true vote count is significantly less than the true winner's count decreasing exponentially with t.

3. Answer:

Preference between the algorithms

1. Algorithm from Question 1:

When to Prefer:

When it's important to release the vote counts for all candidates.

Applicable when voters can vote for multiple candidates.

Useful in scenarios where transparency about overall vote distribution is required.

Considerations:

Higher noise: Due to higher sensitivity, the added noise is significant, leading to less accurate counts.

Utility trade-off: The accuracy of individual counts may be low, especially when n is large.

2. Algorithm from Question 2 (Algorithm 1):

When to Prefer:

When the primary goal is to privately select the winning candidate.

Designed for elections where each voter selects only one candidate.

Provides strong utility guarantees for the selected winner.

Considerations:

Does not provide counts: Only outputs the winning candidate, not the detailed vote counts.

Lower noise: Lower sensitivity allows for less noise, resulting in higher accuracy in winner selection.

3. Exponential mechanism:

When to Prefer:

When it's crucial to select an output that maximizes utility while ensuring differential privacy.

Practical when the number of candidates n is moderate, as the mechanism requires computation over all candidates.

Can be adapted to different utility functions beyond just vote counts.

Considerations:

Computational complexity: May be computationally intensive for large n due to the need to compute and normalize exponentials.

Similar utility guarantees: Achieves similar utility guarantees to Algorithm 1 in winner selection.

Summary

1. Use Algorithm from Question 1 when:

Need to release the noisy counts for all candidates.

Voters can vote for multiple candidates.

Can tolerate higher noise levels in the counts.

2. Use Algorithm from Question 2 (Algorithm 1) when:

The goal is to privately determine the winner with high accuracy.

Each voter selects only one candidate.

Detailed counts are not necessary.

3. Use the Exponential Mechanism when:

Require a mechanism that can handle different utility functions and still provide strong privacy guarantees.

The number of candidates is manageable.

Want a theoretically grounded approach that directly balances utility and privacy.

My some thoughts:

The choice between these algorithms depends on the specific requirements of the election system and the desired outcomes.

Algorithm from Question 1 is suitable for scenarios prioritizing transparency and detailed information at the cost of accuracy.

Algorithm from Question 2 (Algorithm 1) is optimal for accurately determining the winner in a private manner when detailed counts are unnecessary.

The exponential mechanism offers a flexible and theoretically robust approach but may be less practical for large candidate pools due to computational demands.