20.11.2024 ***Adaptive data analysis***

Lecture 3: Bounded description length

Source: Lecture notes by Aaron Roth and Adam Smith

Lecturer: Uri Stemmer

In the previous lecture, we considered the following goal: Design (α, β)-statistically accurate mechanisms for k adaptive queries given a sample of size n, with the objective of minimizing the sample size n as a function of k, α, and β.

We examined a baseline solution obtained through sample splitting. However, this approach does not allow us to answer more than a linear number of queries (i.e., with this approach $k \leq n$).

How can we do better?

Before we begin designing better mechanisms, let us present the following useful claim. In general, we want our guarantees to hold even if the analyst is random. The following claim shows that, without loss of generality, we can assume the analyst is deterministic:

Claim 1: To show that a mechanism M is (α, β)-accurate, it suffices to show that it maintains accuracy against any deterministic analyst. That is, it suffices to show that for any distribution $\mathcal{D}$ and any deterministic analyst A it holds that:

$$\Pr_{\substack{S\sim\mathcal{D}^n \\ \mathrm{AG}_{n,k}(A,S,M)}}[\exists i \text{ s.t. } |a_i - q_i(\mathcal{D})| > \alpha] \leq \beta \qquad ((1))$$

Proof: Suppose that Inequality $((1))$ holds for every distribution $\mathcal{D}$ and every deterministic analyst. Now let A be a randomized analyst and let r denote its random coins. Then,

$$\Pr_{\substack{S\sim\mathcal{D}^n \\ \mathrm{AG}_{n,k}(A,S,M)}}[\exists i \text{ s.t. } |a_i - q_i(\mathcal{D})| > \alpha] = \sum_r \Pr[r] \cdot \Pr_{\substack{S\sim\mathcal{D}^n \\ \mathrm{AG}_{n,k}(A,S,M)}}\left[\exists i \text{ s.t. } |a_i - q_i(\mathcal{D})| > \alpha \;\middle|\; r\right]$$

$$\leq \sum_r \Pr[r] \cdot \beta = \beta$$

q.e.d.

Conclusion: Throughout the course, we may assume without loss of generality that the analyst (or adversary) is deterministic.

Compression

Let's begin discussing another method (beyond sample splitting), called transcript-compression. We will show that if a mechanism is "transcript-compressing" then it guarantees accuracy for adaptive queries (with parameters depending on the quality of the compression it ensures).

Definition 2: A mechanism M is transcript-compressing to $b(n,k)$ bits if for every analyst A there exists a set of transcripts H_A of size $|H_A| \leq 2^{b(n,k)}$ such that for every sample S we have:

$$\Pr\left[AG_{n,k}(A,S,M) \in H_A\right] = 1$$

Note: The set of transcripts H_A can depend on the analyst A. That is, it may be a different set for different analysts.

Another Note: Since we assume (w.l.o.g.) that the analyst is deterministic, it is sufficient to describe a transcript $(q_1, a_1, \ldots, q_k, a_k)$ using just the answers, i.e., $(a_1, \ldots, a_k)$. This is because, when the analyst is deterministic, then every q_i is uniquely determined from $(a_1, \ldots, a_{i-1})$ and from the definition of A.

Theorem 3: Let M be a transcript-compressing mechanism to $b(n,k)$ bits. Then, for every analyst A and every distribution $\mathcal{D}$ it holds that:

$$\Pr_{\substack{S\sim\mathcal{D}^n \\ \mathrm{AG}_{n,k}(A,S,M)}}\left[\exists i \text{ s.t. } |q_i(S) - q_i(\mathcal{D})| > \alpha\right] \leq \beta$$

where

$$\alpha = O\left(\sqrt{\frac{b(n,k) + \ln(k/\beta)}{n}}\right)$$

Proof: Fix an analyst A and a distribution $\mathcal{D}$. Recall that, after fixing A, there exists a set H_A of at most $2^{b(n,k)}$ possible transcripts in the interaction between A and M. Each such transcript contains at most k distinct statistical queries. Hence, once A is fixed, there are at most $w = k \cdot 2^{b(n,k)}$ possible statistical queries that may arise during the execution. Let us denote this set of queries by Q_A. Then, according to a theorem discussed in previous lectures (for the non-adaptive case), it holds that:

$$\Pr_{S\sim\mathcal{D}}\left[\max_{q\in Q_A}|q(S) - q(\mathcal{D})| > \sqrt{\frac{\ln(2w/\beta)}{2n}}\right] \leq \beta$$

In particular,

$$\Pr_{\substack{S\sim\mathcal{D}^n \\ \mathrm{AG}_{n,k}(A,S,M)}}\left[\exists i \text{ s.t. } |q_i(S) - q_i(\mathcal{D})| > \sqrt{\frac{\ln(2w/\beta)}{2n}}\right] \leq \beta$$

q.e.d.

Notes:

1. We are not done yet. All we have shown is that if a mechanism M is transcript-compressing then $q_i(S) \approx q_i(\mathcal{D})$. What we actually want to show is that $a_i \approx q_i(\mathcal{D})$.
2. In particular, if the mechanism simply ignores the queries and always outputs $a_i = 0$, then it allows transcript-compression to 0 bits, and thus $q_i(S) \approx q_i(\mathcal{D})$, but its answers are clearly useless.
3. We are interested in transcript-compressing mechanisms that provide empirically accurate answers.

Definition 4: A mechanism M is $(\boldsymbol{\alpha}, \boldsymbol{\beta})$**-empirically-accurate** for k adaptive queries given a sample of size n if for every sample S of size n and every analyst A it holds that

$$\Pr_{\mathrm{AG}_{n,k}(A,S,M)}[\exists i \text{ s.t. } |a_i - q_i(S)| > \alpha] \leq \beta$$

Theorem 5: Let M be a mechanism such that:

1. M is transcript-compressing to $b(n,k)$ bits.
2. M is (α', β')-empirically-accurate for k adaptive queries given a sample of size n.

Then, for any $\beta'' > 0$ it holds that M is (α, β)-statistically accurate for $\beta = \beta' + \beta''$ and

$$\alpha = \alpha' + O\left(\sqrt{\frac{b(n,k) + \ln(k/\beta'')}{n}}\right)$$

הוכחה:

Proof: Denote $\alpha'' = O\left(\sqrt{\frac{b(n,k)+\ln(k/\beta'')}{n}}\right)$. As M is both (α', β')-empirically-accurate and transcript-compressing to $b(n,k)$ bits, by the union bound,

$$\Pr_{\substack{S\sim\mathcal{D}^n \\ \mathrm{AG}_{n,k}(A,S,M)}}\left[\left\{\exists i \text{ s.t. } |q_i(S) - q_i(\mathcal{D})| > \alpha''\right\} \text{ OR } \left\{\exists i \text{ s.t. } |a_i - q_i(S)| > \alpha'\right\}\right] \leq \beta'' + \beta'$$

That is, with probability at least $1 - \beta'' - \beta'$, for every i it holds that

$$|q_i(S) - q_i(\mathcal{D})| \leq \alpha'' \qquad \text{and} \qquad |a_i - q_i(S)| \leq \alpha'$$

In this case, by the triangle inequality,

$$|a_i - q_i(\mathcal{D})| \leq \alpha'' + \alpha'$$

q.e.d.

In fact, it turns out that the last theorem we proved can be generalized beyond statistical queries to the case of low-sensitivity queries:

Definition 6: Let $q: X^n \to \mathbb{R}$ be a function that maps databases of size n over X to $\mathbb{R}$. The function q has sensitivity c if, for all $x_1, \ldots, x_n \in X$, for all $i \in [n]$, and for all $x_i' \in X$, it holds that:

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

Note: Statistical queries are functions with sensitivity $c = 1/n$

Definition 7: For a query q, which is a function with sensitivity c, and for a sample $S \sim \mathcal{D}^n$, denote the empirical value of q on S by $q(S)$, and denote the "true" value by:

$$q(\mathcal{D}) = \mathbb{E}_{T \sim \mathcal{D}^n}[q(T)]$$

Theorem 8 (Generalization of Theorem 5): Let M be a mechanism that, given a sample of size n, answers queries with sensitivity $1/n$ such that:

1. M allows transcript compression to $b(n, k)$ bits.
2. M is (α', β')-empirically accurate for k adaptive queries given a sample of size n.

Then, for any $\beta'' > 0$, it holds that M is (α, β)-statistically-accurate for $\beta = \beta' + \beta''$ and

$$\alpha = \alpha' + O\left(\sqrt{\frac{b(n,k) + \ln(k/\beta'')}{n}}\right)$$

The proof of this generalization is very similar to the proof of Theorem 5. We will not give the proof here (all that needs to change is that instead of using Chernoff's bound, we use a generalization of Chernoff's bound for low-sensitivity functions, known as McDiarmid's inequality.)

Okay. So, we know that if a mechanism is both transcript-compressing and ensures empirical accuracy, then this implies. How can we construct mechanisms that guarantee these two properties? Let's start with a simple example (not particularly surprising). Later, we will discuss more complex mechanisms that achieve better results.

Definition 9: We define the truncation mechanism, denoted M^b_{Tranc} , as follows: The mechanism takes a sample S as input. Given a query q, it returns the empirical value $q(S)$, truncated to b bits.

Observation 10: The mechanism M^b_{Tranc} is transcript-compressing to $b(n, k) = bk$ bits.

Explanation: We assume (w.l.o.g.) that our analysts are deterministic. Therefore, after fixing an analyst, the transcript can be represented by the list of answers returned by the mechanism. Since each answer is represented by b bits and there are at most k answers in the transcript, the entire transcript can be represented using at most bk bits. Consequently, there are at most 2^{bk} possible transcripts.

Observation 11: The mechanism M^b_{Tranc} is $\left(\frac{1}{2^b}, 0\right)$-empirically accurate.

Explanation: Given a query q, the mechanism returns the empirical value $q(S)$, truncated to b bits. This truncation can distort the answer by at most $1/2^b$.

Conclusion: Let $\beta > 0$ and let $b = \log\left(\sqrt{\frac{n}{k}}\right)$. The truncation mechanism M^b_{Tranc} is (α, β)-statistically accurate, where:

$$\alpha = O\left(\frac{1}{2^b} + \sqrt{\frac{bk + \ln(k/\beta)}{n}}\right) = \tilde{O}\left(\sqrt{\frac{k + \ln(1/\beta)}{n}}\right)$$

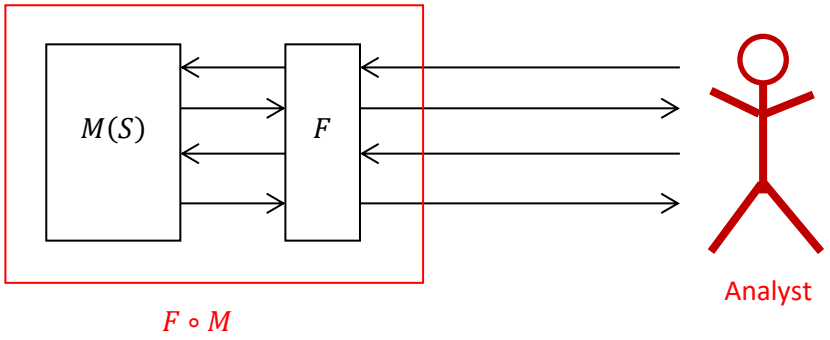
Discussion: The error here scales as $\sqrt{k/n}$. How does this compare to what we got using sample splitting? Did we gain something "conceptually" here?

Next Goal: More serious examples. To get there, we will introduce a few simple/basic transcript-compressing mechanisms, and then design more complex mechanisms that use these basic mechanisms as subroutines. To do this we will need the following properties.

Transcript Compressibility: Composition and Post-processing

Post-processing

Suppose we have a transcript-compressing mechanism M and a deterministic mechanism F that "processes" M's answers (before they are provided to the analyst A) and "processes" A's queries (before they are sent to M). What can we say about the combined mechanism, denoted as $F \circ M$?



Theorem 12: If M is transcript-compressing to b bits, then for any F as described above, $F \circ M$ is also transcript-compressing to b bits.

Remark: Theorem 12 remains valid even if F has a state, meaning the processing it performs at step i depends on everything it has observed in previous steps.

Proof: Fix an analyst A. We need to show that there exists a set of transcripts H of size $|H| \leq 2^b$, such that for every sample S it holds that

$$\Pr\left[AG_{n,k}(A, S, F \circ M) \in H\right] = 1$$

To this end, let us recall the definition of the game $\text{AdaptiveGame}_{n,k}(A, S, F \circ M)$, where we write out the operation of F explicitly:

$\text{AdaptiveGame}_{n,k}(A, S, F \circ M)$

1. The mechanism M gets the sample S (the analyst A does not get S)
2. For $i = 1,2, \ldots, k$:
   - The analyst chooses a query q_i
   - F gets the query q_i and returns a query $\hat{q}_i$. We write this as $\hat{q}_i = F(q_i)$
   - The mechanism M gets the query $\hat{q}_i$ and returns and answer a_i
   - F gets the answer a_i and returns $\hat{a}_i$. We write this as $\hat{a}_i = F(a_i)$
   - The analyst A gets $\hat{a}_i$
3. Return the transcript $T = (q_1, \hat{a}_1, q_2, \hat{a}_2, \ldots, q_k, \hat{a}_k)$

We want to show that the transcript $(q_1, \hat{a}_1, q_2, \hat{a}_2, \ldots, q_k, \hat{a}_k)$ is compressible, i.e., the transcript of the interaction between the analyst A and the mechanism $F \circ M$. As we mentioned, since the analyst is deterministic, it suffices to show this for the part of the transcript containing the answers, i.e., $(\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_k)$.

Convenient Perspective: For the analysis, we can think of F as part of the analyst rather than part of the mechanism (denote this combined analyst as A'). In this view, we consider the transcript $T' = (\hat{q}_1, a_1, \hat{q}_2, a_2, \ldots, \hat{q}_k, a_k)$, which describes the interaction between F and M.

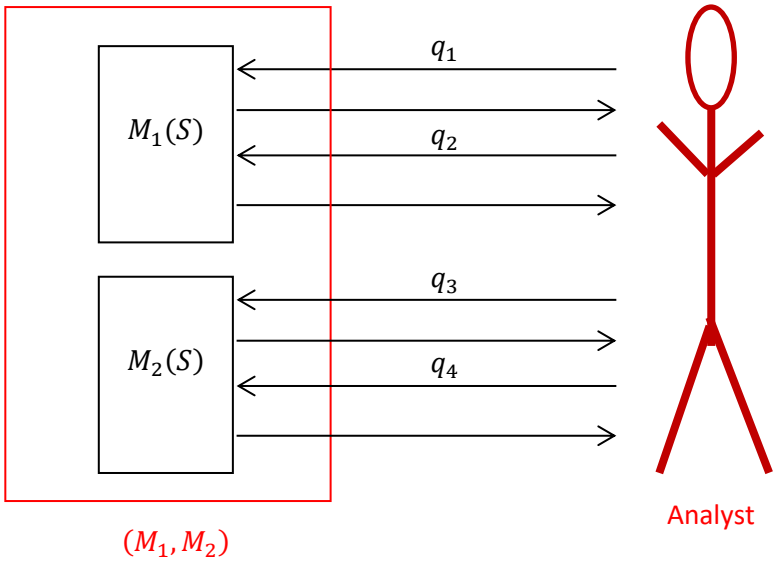
Since M is transcript-compressing to b bits, and since the definition of transcript compression applies to any analyst, we know that in the interaction between M and F, there are at most 2^b possible transcripts. Let this set be denoted by $H_{A'}$, where $|H_{A'}| \leq 2^b$.

Now, since F is deterministic, each transcript in $H_{A'}$ is mapped to exactly one transcript of "translated answers" $(\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_k)$. Therefore, the number of possible "translated answer transcripts" is at most 2^b.

* The number can be smaller if two transcripts from $H_{A'}$ result in the same transcript of "translated answers", but it cannot exceed $|H_{A'}|$.

Composition

Suppose we have two mechanisms M_1, M_2, each of which is transcript-compressing. Consider the mechanism (M_1, M_2), which takes a sample S, feeds it to both M_1 and M_2, then answers k_1 queries using M_1 and subsequently answers k_2 queries using M_2.



Theorem 13: If M_1 is transcript-compressing to $b_1(n, k_1)$ bits and M_2 is transcript-compressing to $b_2(n, k_2)$ bits, then the combined mechanism (M_1, M_2) is transcript-compressing to $b(n, k_1, k_2) = b_1(n, k_1) + b_2(n, k_2)$ bits.

Proof: Fix an analyst A. We need to show that there exists a set of transcripts H of size $|H| \leq 2^{b(n,k_1,k_2)}$ such that for every sample S:

$$\Pr\left[AG_{n,(k_1+k_2)}\left(A, S, (M_1, M_2)\right) \in H\right] = 1$$

To this end, we recall the definition of the game $AG_{n,(k_1+k_2)}\left(A, S, (M_1, M_2)\right)$, where we write out the operation of (M_1, M_2) explicitly:

$\mathbf{AdaptiveGame}_{n,(k_1+k_2)}(A, S, (M_1, M_2))$

1. Each of the mechanisms M_1, M_2 receives the sample S (the analyst A does not get S)
2. For $i = 1,2, \dots, k_1$:
   - The analyst chooses a query q_i
   - The mechanism M_1 gets the query q_i and returns and answer a_i
   - The analyst A gets a_i
3. For $i = k_1 + 1, k_1 + 2, \dots, k_1 + k_2$:
   - The analyst chooses a query q_i
   - The mechanism M_2 gets the query q_i and returns and answer a_i
   - The analyst A gets a_i
4. Return the transcript $T = (q_1, a_1, , \dots, q_{k_1+k_2}, a_{k_1+k_2})$

First, since M_1 is transcript-compressing to $b_1(n, k_1)$ bits, we know there exists a set of transcripts H_A^1 of size $|H_A^1| \leq 2^{b_1(n,k_1)}$, such that for every S we have

$$\Pr[AG_{n,k_1}(A, S, M_1) \in H_A^1] = 1$$

- Let $T_1 = (q_1, a_1, \dots, q_{k_1}, a_{k_1})$ denote the portion of the transcript generated from the interaction between the analyst A and the mechanism M_1
- Denote by A_{T_1} the analyst A in the internal state obtained after completing the interaction with M_1, given that the transcript is T_1.
- Since M_2 is transcript-compressing to $b_2(n, k_2)$ bits, for every T_1 there exists a set $H_{A_{T_1}}^2$ of size $|H_{A_{T_1}}^2| \leq 2^{b_2(n,k_2)}$ such that for every sample S:

$$\Pr\left[AG_{n,k_2}(A_{T_1}, S, M_2) \in H_{A_{T_1}}^2\right] = 1$$

Define:

$$H_A = \left\{(T_1, T_2) \ : \ T_1 \in H_A^1, \ T_2 \in H_{A_{T_1}}^2\right\}$$

We have

$$\Pr[AG_{n,(k_1+k_2)}(A, S, (M_1, M_2)) \in H_A] = 1$$

And in addition,

$$|H_A| \leq \sum_{T_1 \in H_A^1} |H_{A_{T_1}}^2| \leq 2^{b_1(n,k_1)} \cdot 2^{b_2(n,k_2)} = 2^{b_1(n,k_1)+b_2(n,k_2)}$$

q.e.d.

Great. So mechanisms that are transcript-compressing support both **post-processing** and **composition**. How can we use this to design better mechanisms?

Algorithm AboveThreshold

Let's begin with a basic tool that doesn't directly answer queries. Later, we'll use it as a building block for more complex mechanisms.

AboveThreshold$(S, T, q_1, q_2, \dots)$

1. AllDone ← FALSE
2. While not AllDone do:
    a) Accept the next query q_i
    b) Compute $a_i \leftarrow q_i(S)$
    c) if $a_i < T$ then return $\perp$
    d) else
        * return $\top$
        * AllDone ← TRUE

Theorem 14: For any value of T, algorithm AboveThreshold(T) is transcript-compressing to $b(n,k) = \log(k+1)$ bits.

Proof: The sequence of answers given by AboveThreshold is of the form $\perp^i \top$ for $0 \le i \le k-1$, or $\perp^k$. This is a set of $k+1$ strings.

Although this tool seems simple, as we will see later, it is surprisingly powerful. Let's start with a basic use case.

A basic use case for AboveThreshold

We will design a mechanism that, at each step, receives a query q_i (with sensitivity $1/n$) and a "guess" g_i for the value of that query. As long as the guess is close to the query's actual value, the algorithm confirms this and moves to the next round. The first time the guess is incorrect, the algorithm will indicate this and return an approximation of the query's true value.

OneWrongGuess$(S, \eta, (q_1, g_1), (q_2, g_2), \dots)$

1. Start an instance of AboveThreshold on S with threshold η
2. While AboveThreshold has not halted do
    a. Accept the next query (q_i, g_i)
    b. Feed AboveThreshold the query $\hat{q}_i$ defined as $\hat{q}_i(S) = |q_i(S) - g_i|$
    c. if AboveThreshold returns $\perp$ then return the answer $a_i = g_i$
3. Return the answer $a_i = M^b_{\text{Tranc}}(S, q_i)$ for $b = \log(1/\eta)$

Theorem 15: For any $0 < \eta \leq 1$, algorithm OneWrongGuess is $(\eta, 0)$-empirically accurate and transcript-compressing to $b(n,k) = \log(k+1) + \log(1/\eta)$ bits.

Proof:
Consider a "post-processing" mechanism F that transforms the query (q_i, g_i) into a query $\hat{q}_i(S) = |q_i(S) - g_i|$, and replaces an answer $a_i = \perp$ with $a_i = g_i$.

With this notation, the mechanism OneWrongGuess can be represented as

$$\left(F \circ \text{AboveThreshold},\ M^b_{\text{Tranc}}\right)$$

That is, as a composition of $F \circ$ AboveThreshold with M^b_{Tranc}.

Since AboveThreshold is transcript-compressing to $\log(k+1)$ bits, by the post-processing theorem, $F \circ$ AboveThreshold is also transcript-compressing to $\log(k+1)$ bits. Therefore, by the composition theorem, algorithm OneWrongGuess is transcript-compressing to $b(n,k) = \log(k+1) + \log(1/\eta)$ bits.

The claim about empirical accuracy is trivial: we know that M^b_{Tranc} is $(1/2^b, 0)$-empirically accurate, where $1/2^b = \eta$ by our choice of $b = \log(1/\eta)$. In the other queries we answer, we return the value g_i, and by the definition of the AboveThreshold, we know $|g_i - q_i(S)| \leq \eta$.

q.e.d.

Question: What if we want to stop only after the m -th time our guess is incorrect?

Answer: We can simply run OneWrongGuess m times sequentially. Using the composition theorem we proved, we can analyze the compression properties of the resulting algorithm.

Formally,

GuessAndCheck$(S, \eta, m, (q_1, g_1), (q_2, g_2), \ldots)$

1. TimesWrong $\leftarrow 0$
2. While TimesWrong $< m$ do
   a. Start an instance of AboveThreshold on S with threshold η
   b. While AboveThreshold has not halted do
      - Accept the next query (q_i, g_i)
      - Feed AboveThreshold the query $\hat{q}_i(S) = |q_i(S) - g_i|$
      - if AboveThreshold returns $\perp$ then return the answer $a_i = g_i$
   c. Return the answer $a_i = M^b_{\text{Tranc}}(q_i)$ for $b = \log(4/\eta)$
   d. TimesWrong $\leftarrow$ TimesWrong $+ 1$

Theorem 16: For any η, m, algorithm GuessAndCheck(η, m) is $(\eta, 0)$-empirically accurate and transcript-compressing to $b(n,k) = m \cdot (\log(k+1) + \log(1/\eta))$ bits.

Proof: This follows directly from the composition theorem, as GuessAndCheck is a composition of m executions of OneWrongGuess.

Note: In step 2c we run the truncation mechanism with the parameter $b = \log(\mathbf{4}/\eta)$ (instead of $b = \log(\mathbf{1}/\eta)$ as before). This means that when our guess is incorrect, we return an answer with empirical accuracy $\boldsymbol{\eta}/\mathbf{4}$ instead of just $\boldsymbol{\eta}$, as in the previous algorithm. This adjustment will be useful for the next algorithm we present.

Conclusion: Let $m \in \mathbb{N}$ and $\beta > 0$. Denote $= \sqrt{\frac{m}{n}}$. Then, algorithm GuessAndCheck(η, m) is (α, β)-statistically accurate for any sequence of at most k query+guess pairs (q_i, g_i), until the algorithm stops, where each q_i is a query with sensitivity $1/n$, for:

$$\alpha = O\left(\sqrt{\frac{m \cdot \log\left(\frac{kn}{m}\right) + \log\left(\frac{1}{\beta}\right)}{n}}\right)$$

The conclusion follows from Theorem 16 above, combined with the fact that if a mechanism ensures both empirical accuracy and transcript compression, then it is statistically accurate.

Note that to achieve a small error α, the sample size n only needs to grow logarithmically with k (the total number of queries), which is great. However, n still needs to be significantly larger than m (the number of incorrect guesses during execution) to ensure that α remains small…