22.1.2025 ***Adaptive data analysis***

Lecture 11: Multiplicative Weights + Non-iid Sampling

Source: Lecture notes by Aaron Roth and Adam Smith

Lecturer: Uri Stemmer

Theorem 1 (DPMW): There exists an (ϵ,δ)-DP-stable algorithm which, given a database $S \in X^n$, answers k adaptive counting queries with α-empirical-accuracy (w.h.p.), provided

$$n \gtrsim \frac{\sqrt{\log|X| \cdot \log\frac{1}{\delta}} \cdot \log k}{\alpha^2 \varepsilon}$$

(The algorithm runs in $\text{poly}(n, |X|)$ time for each query)

Conclusion 2: (follows from Theorem 1 above + Theorem 5 from Lecture 6)
There exists a computationally inefficient mechanism that answers k adaptively chosen statistical queries to within accuracy α using a sample of size $\boldsymbol{n \gtrsim \frac{\sqrt{\log|X|} \cdot (\log k)^{1.5}}{\alpha^3}}$.

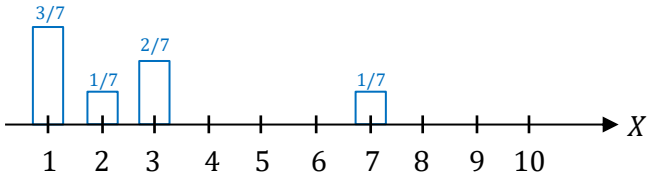
In fact, Theorem 5 from Lecture 6 can be strengthened such that the resulting bound would be linear in $\log k$

This should be compared with the bound of $\boldsymbol{n \approx \frac{\sqrt{k}}{\alpha^2}}$ we got using the (computationally efficient) Laplace mechanism, and with the bound of $\boldsymbol{n \approx \frac{(\log k)^3 \cdot (\log|X|)^2}{\alpha^4}}$ we got using the (inefficient) MedianMechanism.

Proof: It will be convenient for us to think of the database S as a distribution over X:

$$\forall d \in X \quad : \quad S(d) = \frac{|\{x \in S : x = d\}|}{n} = \frac{\text{number of occurrences of } d \text{ in } S}{n}$$

Illustration: If the domain is $X = \{1,2,3,\ldots,10\}$, then for the database $S = (1,1,7,3,1,3,2)$, we obtain the representation



Note that with this notation, for a counting query f, we can write

$$f(S) = \frac{\sum_{x \in S} f(x)}{n} = \frac{\sum_{d \in X} |\{x \in S \ : \ x = d\}| \cdot f(d)}{n} =$$

$$= \sum_{d \in X} \frac{|\{x \in S \ : \ x = d\}|}{n} \cdot f(d) = \sum_{d \in X} S(d) \cdot f(d) = \mathbb{E}_{d \sim S}[f(d)]$$

In the algorithm, we will maintain a "proxy database" $\hat{S}$ (which we will also represent as a distribution over X). When a query arrives, we compare the answer to the query according to S and $\hat{S}$. If the answers are close, we respond based on $\hat{S}$. Otherwise, we update $\hat{S}$ to make it "closer" to S. In the stability analysis, we will show that the number of updates is small, and pay for composition only for the updates.

The DPMW algorithm

(1) Initialize $\hat{S}$ = The uniform distribution over the domain X.

(2) Do at most $\frac{O(\log|X|)}{\alpha^2}$ times (outer loop)

(a) Let $\hat{\alpha} \leftarrow \frac{\alpha}{2} + \mathrm{Lap}\left(\frac{1}{\varepsilon_0 \cdot n}\right)$ (where ε_0 will be set later)

(b) REPEAT: (inner loop)

(i) Get the next query f

(ii) Let $v \leftarrow \mathrm{Lap}\left(\frac{1}{\varepsilon_0 \cdot n}\right)$

(iii) If $\left|f(S) - f\left(\hat{S}\right)\right| + v < \hat{\alpha}$

Then return answer $a = f\left(\hat{S}\right)$ and CONTINUE with the inner loop

Otherwise return answer $a = f(S) + \mathrm{Lap}\left(\frac{1}{\varepsilon_0 \cdot n}\right)$ and BREAK from inner loop

(c) Update $\hat{S}$ as follows:

(i) For $d \in X$ set

$$g(d) = \begin{cases} \hat{S}(d) \cdot \exp\left(\frac{\alpha}{8} \cdot f(d)\right) & , \quad a > f\left(\hat{S}\right) \\ \hat{S}(d) \cdot \exp\left(-\frac{\alpha}{8} \cdot f(d)\right) & , \quad a < f\left(\hat{S}\right) \end{cases}$$

(ii) For $d \in X$ update

$$\hat{S}(d) = \frac{g(d)}{\sum_{\ell \in X} g(\ell)}$$

(d) CONTINUE with the outer loop

<u>We need to show 2 things</u>:

1. *The algorithm is DP-stable – easy*
2. *The returned answers are α-empirically accurate (w.h.p.) – more involved*

Why is the algorithm DP-stable?

In steps (a)+(b) of the outer loop, we execute the **DPAboveThreshold** algorithm (see Lecture 8), and when we exit the inner loop, an additional computation is performed using the Laplace mechanism (see Lecture 4).

Step (c) updates $\hat{S}$ without further access to the input S (beyond what was computed in the previous steps). Therefore, each iteration of the outer loop is $\approx \varepsilon_0$-DP-stable.

By composition over the $\approx \frac{\log|X|}{\alpha^2}$ *repetitions of the outer loop, algorithm DPMW is* $\approx \left(\sqrt{\frac{\log|X|\cdot\log\frac{1}{\delta}}{\alpha^2}}\cdot\varepsilon_0\,,\delta\right)$*-DP-stable. By setting* $\varepsilon_0 \approx \frac{\varepsilon\cdot\alpha}{\sqrt{\log|X|\cdot\log\frac{1}{\delta}}}$ *we get that DPMW is* (ε,δ)*-DP-stable.*

Why does the algorithm preserve empirical accuracy?

Throughout the execution, at most $3k$ *noises are sampled from* $\mathrm{Lap}\left(\frac{1}{\varepsilon_0\cdot n}\right)$*. According to the properties of the Laplace distribution, w.h.p., all these noises (in absolute value) are smaller than* $O\left(\frac{\log k}{\varepsilon_0\cdot n}\right)$*. By asserting that*

$$n \gtrsim \frac{\log k}{\varepsilon_0\cdot\alpha} \approx \frac{\sqrt{\log|X|\cdot\log\frac{1}{\delta}}\cdot\log k}{\alpha^2\varepsilon}$$

we guarantee that all the noises (in absolute value) are smaller than $\boldsymbol{\alpha/8}$ *(w.h.p.). In this case, we conclude that all the answers returned by the algorithm are accurate up to* $\pm\frac{3\alpha}{4}$*. To see this "*$\frac{3\alpha}{4}$*" note that* $|f(S)-f(\hat{S})|+v<\hat{\alpha}$ *translates to* $|f(S)-f(\hat{S})|\pm\frac{\alpha}{8}<\frac{\alpha}{2}\pm\frac{\alpha}{8}$

We still need to show that the algorithm will not halt halfway through the execution (i.e., that the algorithm will not stop before processing k *queries).*

Claim 3: *Assuming all samples from* $\mathrm{Lap}\left(\frac{1}{\varepsilon_0\cdot n}\right)$ *are smaller than* $\alpha/8$ *(in absolute value), the outer loop will execute at most* $O\left(\frac{\log|X|}{\alpha^2}\right)$ *times.*

Proof idea:

Recall that we think of S and $\hat{S}$ as distributions over X. We define a measure of "distance" between distributions, denoted as $\mathrm{KL}(S\|\hat{S})$ (non-negative). At the start of the execution, we have $\mathrm{KL}(S\|\hat{S})\leq\log|X|$. We will show that after each update to $\hat{S}$ (i.e., after each execution of step (c) in the algorithm), the distance between S and $\hat{S}$ decreases by at least $\Omega(\alpha^2)$.

Conclusion: There can be at most $\approx\frac{\log|X|}{\alpha^2}$ update steps.

Definition: For distributions $S,\hat{S}$ over X, we define the following measure (known as the **Kullback–Leibler divergence**):

$$\mathrm{KL}(S\|\hat{S})=\sum_{d\in X}S(d)\cdot\log\left(\frac{S(d)}{\hat{S}(d)}\right)$$

Fact 1: *For any pair of distributions* $S, \hat{S}$ over X it holds that $\mathrm{KL}(S\|\hat{S}) \geq 0$.

Proof: *This follows from the log-sum inequality, which states that if* $a_1, \ldots, a_n, b_1, \ldots, b_n$ *are non-negative numbers, then*:

$$\sum_i a_i \cdot \log\left(\frac{a_i}{b_i}\right) \geq \left(\sum_i a_i\right) \cdot \log\left(\frac{\sum_i a_i}{\sum_i b_i}\right)$$

Indeed, using this inequality, we obtain

$$\mathrm{KL}(S\|\hat{S}) = \sum_{d\in X} S(d) \cdot \log\left(\frac{S(d)}{\hat{S}(d)}\right) \geq \left(\sum_{d\in X} S(d)\right) \cdot \log\left(\frac{\sum_{d\in X} S(d)}{\sum_{d\in X} \hat{S}(d)}\right) = 1 \cdot \log\left(\frac{1}{1}\right) = 0$$

Here we used the assumption that $S, \hat{S}$ are probability distributions and hence $\sum_{d\in X} S(d) = \sum_{d\in X} \hat{S}(d) = 1$

Fact 2: If $\hat{S}$ is the uniform distribution over X then for any distribution S it holds that

$$\mathrm{KL}(S\|\hat{S}) \leq \log|X|$$

Proof:

$$\mathrm{KL}(S\|\hat{S}) = \sum_{d\in X} S(d) \cdot \log\left(\frac{S(d)}{\hat{S}(d)}\right) = \sum_{d\in X} S(d) \cdot \log\left(|X| \cdot S(d)\right) =$$

$$= \underbrace{\sum_{d\in X} S(d) \cdot \log|X|}_{=\log|X|} + \underbrace{\sum_{d\in X} S(d) \cdot \underbrace{\log\left(S(d)\right)}_{\leq 0}}_{\leq 0} \leq \log|X|$$

Proof of Claim 3:

Suppose we performed an update in step (c) and transitioned from $\hat{S}$ to $\hat{S}'$.
Our goal is to show that

$$\underbrace{\mathrm{KL}(S\|\hat{S}) - \mathrm{KL}\left(S\|\hat{S}'\right)}_{\text{This is the amount by which the "distance" to S decreases. We want to show that this is big.}} \gtrsim \alpha^2$$

The update in step (c) is performed in one of two ways (depending on whether $a > f(\hat{S})$ or not). Let us assume that $a > f(\hat{S})$ (the second case is symmetric), and calculate:

$$\mathrm{KL}(S\|\hat{S}) - \mathrm{KL}\left(S\|\hat{S}'\right) = \sum_{d\in X} S(d) \cdot \log\left(\frac{S(d)}{\hat{S}(d)}\right) - \sum_{d\in X} S(d) \cdot \log\left(\frac{S(d)}{\hat{S}'(d)}\right)$$

$$= \sum_{d\in X} S(d) \cdot \log\left(\frac{\hat{S}'(d)}{\hat{S}(d)}\right) = \sum_{d\in X} S(d) \cdot \log\left(\frac{g(d)/\sum_{\ell\in X} g(\ell)}{\hat{S}(d)}\right) =$$

$$= \left[\sum_{d\in X} S(d)\cdot \log\left(\frac{g(d)}{\hat{S}(d)}\right)\right] - \log\left(\sum_{\ell\in X} g(\ell)\right)$$

$$= \left[\sum_{d\in X} S(d)\cdot \log\left(\exp\left(\frac{\alpha}{8}\cdot f(d)\right)\right)\right] - \log\left(\sum_{\ell\in X} \hat{S}(\ell)\cdot \exp\left(\frac{\alpha}{8}\cdot f(\ell)\right)\right)$$

$$= \left[\sum_{d\in X} S(d)\cdot \frac{\alpha}{8}\cdot f(d)\right] - \log\left(\sum_{\ell\in X} \hat{S}(\ell)\cdot \exp\left(\frac{\alpha}{8}\cdot f(\ell)\right)\right)$$

$$= \frac{\alpha}{8}\cdot f(S) - \log\left(\sum_{\ell\in X} \hat{S}(\ell)\cdot \exp\left(\frac{\alpha}{8}\cdot f(\ell)\right)\right) = ((1))$$

Recall that for all $y \le 1$, it holds that $e^y \le 1 + y + y^2$. Therefore,

$$\exp\left(\frac{\alpha}{8}\cdot f(\ell)\right) \le 1 + \frac{\alpha}{8}\cdot f(\ell) + \frac{\alpha^2}{64}\cdot \underbrace{f^2(\ell)}_{\le 1}$$

And so,

$$((1)) \ge \frac{\alpha}{8}\cdot f(S) - \log\left(\sum_{\ell\in X} \hat{S}(\ell)\cdot\left(1 + \frac{\alpha}{8}\cdot f(\ell) + \frac{\alpha^2}{64}\right)\right)$$

$$= \frac{\alpha}{8}\cdot f(S) - \underbrace{\log\left(1 + \frac{\alpha}{8}\cdot f(\hat{S}) + \frac{\alpha^2}{64}\right)}_{\forall y\ :\ \log(1+y)\le y}$$

$$\ge \frac{\alpha}{8}\cdot \underbrace{\left(f(S) - f(\hat{S})\right)}_{\substack{\ge \alpha/4 \\ \text{(explained next)}}} - \frac{\alpha^2}{64} \ge \frac{\alpha^2}{64}$$

Why is it the case that $f(S) - f(\hat{S}) \ge \alpha/4$?

We assume that all the noises throughout the execution (in absolute value) are at most $\alpha/8$. Therefore, when an update occurs, it holds that

$$\left|f(S) - f(\hat{S})\right| + v > \hat{\alpha}$$

$$\Downarrow$$

$$\left|f(S) - f(\hat{S})\right| > \frac{\alpha}{2} - \frac{\alpha}{8} - \frac{\alpha}{8} = \frac{\alpha}{4}$$

Additionally, we are analyzing the case where

$$f(S)+\frac{\alpha}{8} \underbrace{\geq}_{\substack{\text{By the algorithm}\\ a=f(S)+\text{Noise}\\ \text{and the noise is bounded by } \frac{\alpha}{8}}} a \underbrace{>}_{\substack{\text{the case we}\\ \text{are analyzing}}} f(\hat{S})$$

$$\Downarrow$$

$$f(\hat{S})-f(S)\leq\frac{\alpha}{8}$$

So we have both $|f(S)-f(\hat{S})|>\alpha/4$ *and* $f(\hat{S})-f(S)\leq\alpha/8$.

Therefore :

$$f(S)-f(\hat{S})>\frac{\alpha}{4}$$

q.e.d.

Answering adaptive queries when there are dependencies in the sample

Until now, when we discussed answering adaptive queries in the course, we assumed that our data was sampled i.i.d. In real life, in many cases, the data we collect is not sampled i.i.d. For example, a surveyor arrives at a particular house and conducts the survey on each of the house's residents. What can we say without the assumption that the data is sampled i.i.d?.

Let us try to formalize this question. Our AdaptiveGame does not change:

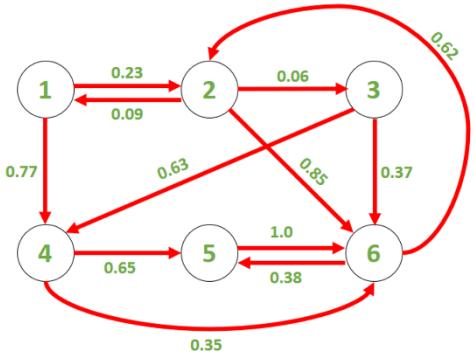
$\mathbf{AdaptiveGame}_{n,k}(A,S,M)$

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 (assume that $q_i\in Q$ for a family of possible queries Q)
   - The mechanism M gets the query q_i and returns and answer a_i
   - The analyst A gets a_i
3. Return the transcript $T=(q_1,a_1,q_2,a_2,\ldots,q_k,a_k)$ of the interaction between A and M

Previously, we assumed that the sample S contains n i.i.d. samples from some distribution $\mathcal{D}$ over a domain X. We still want to assume that the sample S contains n samples, but now dependencies within the sample are possible. Now we will assume that the target distribution, which we will call $\mathcal{P}$, is a distribution over n-tuples. That is, $\mathcal{P}$ is a distribution over X^n.

Note that this generalizes our previous representation, as $\mathcal{P}$ might indeed be a product distribution where each of the n entries is sampled i.i.d. from some distribution $\mathcal{D}$ over X.

Example: $\mathcal{P}$ might be defined by a Markov chain. For instance, to sample from $\mathcal{P}$, we start in state 1 and move to the next state randomly according to the transition probabilities. We repeat this process for n steps, and the result of our sampling is the sequence of states we visited



Another example: $\mathcal{P}$ might be defined as follows. To sample from $\mathcal{P}$, we start by sampling $x_1 \in [0,100]$ uniformly. Then, for $i = 2, \dots, n$, we sample x_i uniformly from $\left[\frac{1}{i-1}\sum_{j=1}^{i-1} x_j, \frac{2}{i-1}\sum_{j=1}^{i-1} x_j\right]$.

The takeaway here is that the target distribution $\mathcal{P}$ is now a distribution over n-tuples rather than over singletons.

If that is the case, then given a query, how do we define its value over the distribution $\mathcal{P}$? In other words, given a query, what would we want to return?

Definition: For a distribution $\mathcal{P}$ over X^n and a statistical query $q: X \to [0,1]$, we denote

$$q(\mathcal{P}) = \mathbb{E}_{S\sim\mathcal{P}}[q(S)] = \mathbb{E}_{S\sim\mathcal{P}}\left[\frac{1}{|S|}\sum_{x\in S} q(x)\right]$$

Note: We could also discuss queries that operate on n-tuples, such as low-sensitivity queries or other types of queries. Here, we will stick to statistical queries for simplicity.

Proposal: Perhaps we should require the same thing we demanded in the i.i.d. case, but under the assumption that the mechanism receives a sample $S \sim \mathcal{P}$. That is,

Proposed definition: A mechanism M is **(α,β)-statistically-accurate (**with dependencies) for k adaptive queries over X if, for every distribution $\mathcal{P}$ over X^n and every analyst A, it holds that

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

Is this a good definition?

This is a definition that might be impossible to satisfy. For example, $\mathcal{P}$ could be one of the following two distributions:

$\mathcal{P}_1$	$\mathcal{P}_0$
• *W.p.* 0.5 *return* $\left(\frac{1}{2},\frac{1}{2},\dots,\frac{1}{2}\right)$ • *W.p.* 0.5 *return* $(1,1,\dots,1)$	• *W.p.* 0.5 *return* $\left(\frac{1}{2},\frac{1}{2},\dots,\frac{1}{2}\right)$ • *W.p.* 0.5 *return* $(0,0,,\dots,0)$

What would we respond when holding the sample $\left(\frac{1}{2},\frac{1}{2},\dots,\frac{1}{2}\right)$ *and receiving the query* $q(x) = x$?

Conclusion: We must restrict the problem somehow.

Two main approaches:

1. **Explicitly limit dependencies within the sample.** Intuitively, in the last example, there were very "strong" dependencies within the sample. Under an "appropriate" restriction of the dependencies, and assuming the sample indeed comes from a distribution that meets these restrictions, it is possible to design mechanisms that answer adaptive queries.

2. **Compare the two worlds** – the adaptive world and the non-adaptive world. We aim to answer adaptive queries in a manner similar to what can be achieved in the non-adaptive world. Note that the last example cannot be solved even in the non-adaptive world, so in a sense, it is "unfair" to attempt solving it in the adaptive world. "If something cannot be solved in the non-adaptive world, how can we solve it in the adaptive world?"

Today, we will focus on **approach 2**.

Note: Approach 2 also effectively restricts the target distribution, but in a less explicit way and not through a specific measure of dependencies.

Definition: Let $\mathcal{P}$ be a distribution over X^n, and let $q: X \to [0,1]$ be a statistical query. We say that q is **(α,β)-concentrated with respect to $\mathcal{P}$** if

$$\Pr_{S\sim\mathcal{P}}[|q(S) - q(\mathcal{P})| > \alpha] < \beta$$

In other words, a query q is concentrated with respect to the distribution $\mathcal{P}$ if, when sampling $S \sim \mathcal{P}$, it holds with high probability (over the draw of S) that the empirical average is close to the expectation. Intuitively, in general, for queries that are not concentrated, we do not have a good way to answer them even in the non-adaptive world. Therefore, it makes sense to focus on concentrated queries.

Example: If $\mathcal{P}$ is a product distribution, i.e., $\mathcal{P} = \mathcal{D}^n$, then every statistical query q is concentrated with respect to $\mathcal{P}$ by the Hoeffding bound.

Another example (lazy random walk): If $\mathcal{P}$ is the following distribution over X^n, then every statistical query q is concentrated with respect to $\mathcal{P}$.

- Let $\mathcal{D}$ be some distribution over X.
- To sample from $\mathcal{P}$, perform the following process:
    1. Sample $x_1 \sim \mathcal{D}$.
    2. For $i = 2, \dots, n$:
        - With probability ½, set $x_i = x_{i-1}$.
        - With probability ½, sample $x_i \sim \mathcal{D}$.
- Return $(x_1, x_2, \dots, x_n)$.

Definition 4: A mechanism M is $(\alpha, \hat{\alpha}, \beta, \hat{\beta})$**-statistically-accurate** for k concentrated adaptive queries over X if, for every distribution $\mathcal{P}$ over X^n and every analyst A asking $(\hat{\alpha}, \hat{\beta})$-concentrated queries with respect to $\mathcal{P}$, it holds that

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

How can we design mechanisms that satisfy Definition 4?

Note: Observe that now it is not even clear if we can use the Sample Splitting method, since due to the dependencies, Hoeffding's bound might not apply. For example:

- When sampling $S \sim \mathcal{P}$, perhaps all of S is encoded in the low-order bits of elements in S, such that an (accurate) empirical average over part of S reveals the entirety of S.
- Another example: Perhaps the distribution $\mathcal{P}$ generates t duplicated blocks, i.e., it samples one block and then replicates it t times.

Today, we will see how tools we studied for **transcript compression** can be applied even when there are dependencies within the sample.

Recall:

Definition: 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:

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

We will restrict the analyst to ask only queries that are concentrated with respect to the target distribution.

Theorem 5: Let M be a transcript-compressing mechanism to $b(n,k)$ bits. Then, for every distribution $\mathcal{P}$ and every analyst A asking $(\hat{\alpha}, \hat{\beta})$-concentrated queries with respect to $\mathcal{P}$ it holds that:

$$\Pr_{\substack{S\sim\mathcal{P} \\ \mathrm{AG}_{n,k}(A,S,M)}} \left[\exists i \text{ s.t. } |q_i(S) - q_i(\mathcal{P})| > \hat{\alpha}\right] \leq \hat{\beta} \cdot k \cdot 2^{b(n,k)}$$

Proof: Fix a distribution $\mathcal{P}$ and fix an analyst A. Recall that after fixing A there is a set H_A of at most $2^{b(n,k)}$ possible transcripts between A and M. In every such transcript there are at most k queries, and so there are at most $2^{b(n,k)} \cdot k$ possible queries throughout the execution. Let us denote this set of all possible queries as Q_A. As these queries are concentrated, by the union bound it holds that

$$\Pr_{S\sim\mathcal{P}} \left[\max_{q\in Q_A} |q(S) - q(\mathcal{P})| > \hat{\alpha}\right] \leq \hat{\beta} \cdot k \cdot 2^{b(n,k)}$$

And in particular,

$$\Pr_{\substack{S\sim\mathcal{P} \\ \mathrm{AG}_{n,k}(A,S,M)}} \left[\exists i \text{ s.t. } |q_i(S) - q_i(\mathcal{P})| > \hat{\alpha}\right] \leq \hat{\beta} \cdot k \cdot 2^{b(n,k)}$$

q.e.d.

Theorem 6 (analogous to the theorem we saw for the i.i.d. case):
Let M be a mechanism such that:

- M is **transcript compressing** to $b(n,k)$ bits.
- M is (α',β')-empirically accurate for k adaptive queries.

Then, for any $\hat{\alpha},\hat{\beta}$, it holds that M is $(\alpha,\hat{\alpha},\beta,\hat{\beta})$-statistically accurate for

$$\beta = \beta' + \hat{\beta}\cdot k\cdot 2^{b(n,k)} \qquad \text{and} \qquad \alpha = \alpha' + \hat{\alpha}$$

The proof is nearly identical to the corresponding proof for the i.i.d. case (just the triangle inequality...)

Reminder:

***There exists a mechanism (the MedianMechanism) satisfying*:**

a. *Allows transcript compression to* $b(n,k) = \tilde{O}\left(\sqrt{n}\cdot\log|X|\cdot(\log k)^{3/2}\right)$
b. *Achieves* $(\alpha',0)$*- empirical accuracy for* $\alpha' = \left(\frac{\ln(4k)}{2n}\right)^{1/4}$

Conclusion 7: For every $\hat{\alpha},\hat{\beta}$ there exists a mechanism that is $(\alpha,\hat{\alpha},\beta,\hat{\beta})$-statistically accurate for

$$\beta = 0 + \hat{\beta}\cdot k\cdot 2^{\tilde{O}\left(\sqrt{n}\cdot\log|X|\cdot(\log k)^{3/2}\right)} \qquad \text{and} \qquad \alpha = \left(\frac{\ln(4k)}{2n}\right)^{1/4} + \hat{\alpha}$$

Discussion about Conclusion 7:

The main penalty we pay here is that the error probability β "explodes". This means that, to obtain something useful, we need to assume that the queries asked by the analyst are very concentrated with respect to the target distribution. In particular, we need $\hat{\beta} \ll 2^{-\sqrt{n}}$

This holds when the target distribution is a product distribution: if $\mathcal{P} = \mathcal{D}^n$, then by Hoeffding's inequality, for any statistical query $q: X \to [0,1]$ and any $\hat{\beta} > 0$, it holds that

$$\Pr_{S\sim\mathcal{P}}\left[|q(S) - q(\mathcal{P})| > \sqrt{\frac{\ln(2/\hat{\beta})}{2n}}\right] \le \hat{\beta} \qquad ((1))$$

That is, in the iid case, every statistical query is $(\hat{\alpha},\hat{\beta})$-concentrated for $\hat{\alpha} = \sqrt{\frac{\ln(2/\hat{\beta})}{2n}}$.

To obtain something meaningful from Conclusion 7, we want to ensure that $\hat{\beta}$ is sufficiently small. Specifically, we require

$$1 \gg \hat{\beta}\cdot k\cdot 2^{\tilde{O}\left(\sqrt{n}\cdot\log|X|\cdot(\log k)^{3/2}\right)}$$

Which roughly means that

$$\hat{\beta} \ll \frac{1}{k} \cdot 2^{-\sqrt{n} \cdot \log|X| \cdot (\log k)^{3/2}}$$

Plugging this into $\big((1)\big)$ *we get that every statistical query is concentrated with*

$$\hat{\alpha} \approx \sqrt{\frac{\ln(2k) + \sqrt{n} \cdot \log|X| \cdot (\log k)^{\frac{3}{2}}}{2n}} \approx \sqrt{\frac{\log|X| \cdot (\log k)^{\frac{3}{2}}}{\sqrt{n}}}$$

Which by Conclusion 7 gives accuracy of

$$\alpha \approx \left(\frac{\ln(4k)}{2n}\right)^{1/4} + \sqrt{\frac{\log|X| \cdot (\log k)^{\frac{3}{2}}}{\sqrt{n}}}$$

Note that the error decreases with n*.*

Notes:

- This is exactly what we obtained for the **MedianMechanism** when we analyzed it directly for the i.i.d. case. In other words, we generalized the analysis without losing anything (when degenerating the result back to the i.i.d. case).

- While this calculation was specifically for product distributions $\mathcal{P} = \mathcal{D}^n$, the only thing we used here was the assumption that the queries asked by the analyst are highly concentrated. The conclusion is that as long as the queries asked by the analyst are highly concentrated, we can guarantee non-trivial error bounds even when there are dependencies in the sample.