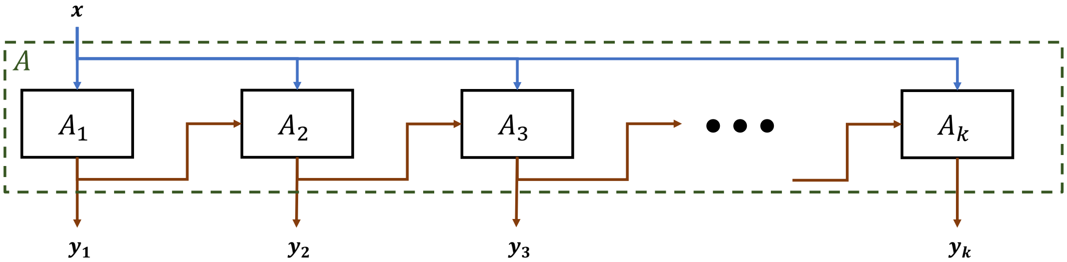
# Algorithms for Private Data Analysis — Spring 2021
# Lecture 11: Advanced Composition

**Steven Wu**

## 1 Advanced Composition of Approximate DP

In this lecture, we show that $(\varepsilon, \delta)$-differential privacy enables a stronger form of composition, in which the $\varepsilon$ parameter increases only with the *square root* of the number of stages of the composition.

Consider an algorithm $A$ that consists of the adaptive composition of $k$ algorithms, each of which is $(\varepsilon, \delta)$-DP:



Previously, we argued that if each individual algorithm is $(\varepsilon, 0)$-DP, then the composition of all $k$ algorithms is (at worst) $(k\varepsilon, 0)$-DP. That is the best one can hope to prove for $(\varepsilon, 0)$-DP, but the relaxation to $(\varepsilon, \delta)$ gives us a different type of guarantee:

**Theorem 1.1** (Strong Composition)**.** *For all $\varepsilon, \delta \geq 0$ and $\delta' > 0$, the* adaptive *composition of $k$ algorithms, each of which is $(\varepsilon, \delta)$-differentially private, is $(\tilde{\varepsilon}, \tilde{\delta})$-differentially private where*

$$\tilde{\varepsilon} = \varepsilon\sqrt{2k\ln(1/\delta')} + k\varepsilon\frac{e^{\varepsilon}-1}{e^{\varepsilon}+1} \quad \textit{and} \quad \tilde{\delta} = k\delta + \delta'. \tag{1}$$

Let's get a feeling for the asymptotics here. When $\varepsilon$ is not too big (say, at most 1), the quantity $\frac{e^{\varepsilon}-1}{e^{\varepsilon}+1}$ is close to $\varepsilon/2$, so the final privacy parameter $\tilde{\varepsilon}$ is $\Theta(\varepsilon\sqrt{k\ln(1/\delta)} + \varepsilon^2 k)$ if we take $\delta' = \delta$. Suppose we want this final privacy guarantee to be at most 1, then we need $\varepsilon^2 k < 1$. In that range, we have $\varepsilon\sqrt{k} > \varepsilon^2 k$, so

$$\tilde{\varepsilon} = \Theta\left(\varepsilon\sqrt{k\ln(1/\delta)}\right) \qquad \text{when } \varepsilon < 1/\sqrt{k}.$$

Contrast this with so-called *basic composition* (from Lecture 9), which shows that the adaptive composition of $k$ mechanisms that are $(\varepsilon, \delta)$-DP is $(k\varepsilon, k\delta)$-DP. When $k > \ln(1/\delta)$, strong compositon provides a much tighter bound (see Figure 1 for an example). This is crucial when we analyze iterative algorithms that have many stages, as with the differentially private gradient descent methods we will see later.

For example, consider the task of approximating a set of $d$ count queries. Absent a special relationship between the queries, the global $\ell_1$ sensitivity of the vector of counts is $d$ and so the Laplace mechanism adds noise $\Theta(d/\varepsilon)$ to each query's answer. The Gaussian mechanism from last lecture would add noise of expected magnitude only $\Theta(\sqrt{d\ln(1/\delta)}/\varepsilon)$ because the $\ell_2$ sensitivity of the vector is $\sqrt{d}$.

However, we can alternately view the Laplace mechanism on the whole vector as the composition of $d$ separate instances of the Laplace mechanism—one for each query. If we ensure each one is $(\varepsilon', 0)$-DP, then strong composition implies that the whole algorithm is $(\varepsilon, \delta)$-DP for $\varepsilon = \Theta(\varepsilon'\sqrt{k\ln(1/\delta)}$. Setting

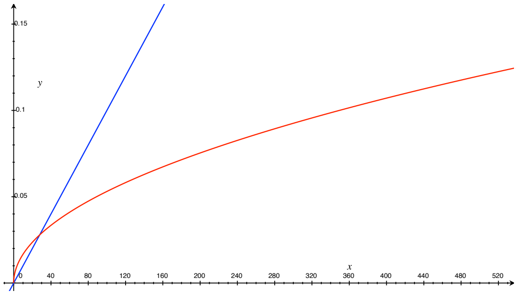


Figure 1: Bounds on the privacy parameter obtained for the composition of $k$ mechanisms, each of which is $(\varepsilon, 0)$-DP for $\varepsilon = 0.01$. The horizontal axis represents the number $k$ of mechanisms. The blue (straight) curve shows the bound $k\varepsilon$ given by basic composition, while the red curve shows the value $\tilde{\varepsilon}$ given by Theorem 1.1 with $\delta' = 10^{-6}$.

$\varepsilon' = \frac{\varepsilon}{\sqrt{k \ln(1/\delta)}}$, we see that the Laplace mechanism satisfies $(\varepsilon, \delta)$differential privacy with a smaller amount of noise—the same $\Theta(\sqrt{d \ln(1/\delta)}/\varepsilon)$ bound we get from the Gaussian mechanism!

**Quantitatively Tighter Bounds** The bound in Theorem 1.1 provides clear asymptotics, but is not always tight. First, we'll see from the proof that the dominant term in the bound on $\tilde{\varepsilon}$ is actually a generic bound on the tails of the binomial distribution; plugging in exact bounds can improve the constant terms.

There are also now many results that yield tighter bounds for the composition of specific mechanisms or classes of mechanisms. These have proven crucial for understanding algorithms with many stages of a particular form, such as stochastic gradient descent (discussed next lecture). For now, though, we will try to see how to prove the simple, general bound of Theorem 1.1.

## 2 Privacy Loss as a Random Variable

Given a randomized algorithm $A$ and two possible inputs $\mathbf{x}$ and $\mathbf{x}'$, define the privacy loss on output $y$ to be the "log-odds ratio", that is, the log of the ratio of the likelihoods of $y$ under $\mathbf{x}$ and $\mathbf{x}'$:

$$I_{\mathbf{x},\mathbf{x}'}(y) \stackrel{\text{def}}{=} \ln\left(\frac{\mathbb{P}\left(A(\mathbf{x}) = y\right)}{\mathbb{P}\left(A(\mathbf{x}') = y\right)}\right). \tag{2}$$

Last lecture, we showed (Lemma 1.4) that if, for every pair of neighboring data sets $\mathbf{x}, \mathbf{x}'$,

$$\mathop{\mathbb{P}}_{Y \leftarrow A(\mathbf{x})}\left(I_{\mathbf{x},\mathbf{x}'}(Y) > \varepsilon\right) \leq \delta\,,$$

then the mechanism $A$ is $(\varepsilon, \delta)$-DP.

Now when $A$ consists of the adaptive composition of $k$ mechanisms, we can write the output as a sequence $y = (y_1, y_2, \ldots, y_k)$. We do not want to assume anything about the way that the $j$-th algorithm $A_j$ is chosen based on $y_1, y_2, \ldots, y_{j-1}$. Somewhat surprisingly, we don't have to! We can break up the probability of seeing the sequence $y$ as a product

$$\mathbb{P}\left(A(\mathbf{x}) = y_1, \ldots, y_k\right) = \mathbb{P}\left(A_1(\mathbf{x}) = y_1\right) \times \mathbb{P}\left(A_2(\mathbf{x}, y_1) = y_2\right) \times \cdots \times \mathbb{P}\left(A_k(\mathbf{x}, y_1, \ldots, y_{k-1}) = y_k\right),$$

which allows us to write the privacy loss as a sum:

$$I_{\mathbf{x},\mathbf{x}'}(y_1,..,y_k) = \sum_{j=1}^{k} \ln\left(\frac{\mathbb{P}\left(A_j(\mathbf{x},y_1,...,y_{j-1}) = y_j\right)}{\mathbb{P}\left(A_j(\mathbf{x}',y_1,...,y_{j-1}) = y_j\right)}\right). \tag{3}$$

The important observation is that in each term of this sum, we are conditioning on the same previous outputs $y_1,...,y_{j-1}$ in the numerator and denominator. Regardless of how $A_j$ is chosen, we are comparing outputs of the same algorithm $A_j$ on both outputs.

Basic composition for $(\varepsilon, 0)$-DP follows from the fact that for such mechanisms each term in the sum (3) is at most $\varepsilon$, so the sum is at most $k\varepsilon$.

To prove the strong composition theorem for $(\varepsilon, \delta)$-DP, we want to take advantage of the fact that there is some cancelation in this sum. We know (roughly) that each term is contained in the interval $[-\varepsilon, \varepsilon]$ with high probability. But it turns out that their average is generally at most $\varepsilon^2$. When many of them are added, that is the behavior which dominates.

## 2.1 Privacy Loss Distributions for Some Representative Mechanisms

To get a sense of that, we can compute this privacy loss for a few example mechanisms, and how it is distributed.

**Gaussian Noise** suppose each $A_j$ is an instance of the Gaussian mechanism from last lecture. The proof of Theorem 2.1 shows that the log-odds ratio is itself normally distributed, namely when $Y$ is the output of the algorithm under data set $\mathbf{x}$, we have $I_{\mathbf{x},\mathbf{x}'}(Y) \sim N\left(\frac{\Delta^2}{2\sigma^2}, \frac{\Delta^2}{\sigma^2}\right)$. We chose $\sigma = \Delta\sqrt{2\ln(1/\delta)}/\varepsilon$, so the privacy loss for this mechanism has expectation $\varepsilon^2 \cdot \frac{1}{4\ln(1/\delta)}$.

**Randomized Response** Let's look at the example of randomized response from Lectures 1 and 4. Each input bit $x_i$ is randomized with a value

$$Y_i = \begin{cases} x_i & \text{w.p. } \frac{e^\varepsilon}{e^\varepsilon+1}, \\ 1 - x_i & \text{w.p. } \frac{1}{e^\varepsilon+1}. \end{cases}$$

For every two neighboring datasets $\mathbf{x}, \mathbf{x}'$, the privacy loss $I_{\mathbf{x},\mathbf{x}'}(y)$ is therefore $\varepsilon$ with probability $\frac{e^\varepsilon}{e^\varepsilon+1}$, and $-\varepsilon$ with probability $\frac{1}{e^\varepsilon+1}$. It's expectation is $\varepsilon \cdot \frac{e^\varepsilon-1}{e^\varepsilon+1} = \Theta(\varepsilon^2)$—again, we see the same scaling.

**Name and Shame** Recall the name and shame algorithm $NS_\delta$ from Lecture 5, which outputs each person's raw data with probability $\delta$. If data sets $\mathbf{x}, \mathbf{x}'$ differ in person $i$'s data, the privacy loss is $+\infty$ when person $i$'s data is released, and 0 when it is not. The expectation of this privacy loss is $\infty$, but only due to the small probability event in which there is a catastrophic failure of secrecy.

We'll see below that these three behaviors are representative—every $(\varepsilon, \delta)$ differentially private algorithm has privacy loss that is roughly $\varepsilon^2$ in expectation, as long as we first set aside some event of probability at most $\delta$.

**Exercise 2.1.** What is the distribution of the privacy loss $I_{\mathbf{x},\mathbf{x}'}(Y)$ when $A$ is the Laplace mechanism in one dimension? Show that its expectation is $\Theta(\varepsilon^2)$.

# 3 Proving Strong Composition

## 3.1 The Simulation Lemma: Reducing to Leaky Randomized Response

To get a handle on the privacy loss, we'll actually show that once we fix two neighboring data sets, every $(\varepsilon, \delta)$-DP algorithm's behavior is captured by a very simple "leaky" variant of randomized response.

If $X$ and $Y$ are random variables taking values in the same set (and with probabilities defined for the same collection of events), we say $X \approx_{\varepsilon,\delta} Y$ if for every event $E$: $P_X(E) \leq e^{\varepsilon} P_Y(E) + \delta$ and $P_Y(E) \leq e^{\varepsilon} P_X(E) + \delta$.

We would like to characterize this relation in simpler terms. As a starting point, let's try to imagine the simplest pair of random variables that satisfies the relationship. It seems like we need one type of outcome to capture the $\delta$ additive difference in probabilities, and another type that captures the $e^{\varepsilon}$ multiplicative change. Consider the following two special random variables, $U$ and $V$, taking values in the set $\{0, 1, \text{“I am U”}, \text{“I am V”}\}$ with the probabilities

| Outcome | $P_U$ | $P_V$ |
|---|---|---|
| 0 | $\frac{e^{\varepsilon}(1-\delta)}{e^{\varepsilon}+1}$ | $\frac{1-\delta}{e^{\varepsilon}+1}$ |
| 1 | $\frac{1-\delta}{e^{\varepsilon}+1}$ | $\frac{e^{\varepsilon}(1-\delta)}{e^{\varepsilon}+1}$ |
| “I am U" | $\delta$ | 0 |
| “I am V" | 0 | $\delta$ |

**Lemma 3.1** (Simulation Lemma for $(\varepsilon, \delta)$-DP). *For every pair of random variables $X, Y$ such that $X \approx_{\varepsilon,\delta} Y$, there exists a randomized map $F$ such that $F(U) \sim X$ and $F(V) \sim Y$.*

**Exercise 3.2.** Prove the Simulation Lemma. We provide the following pictorial hint:

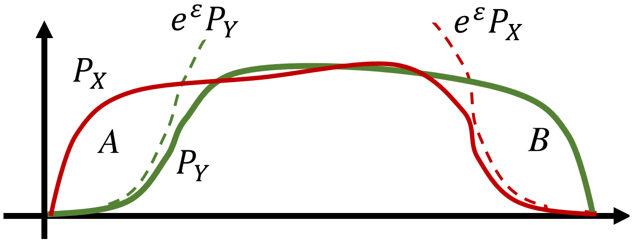


Figure 2: The “proof" of Lemma 3.1

It is ok to assume that $X$ and $Y$ take values in a discrete set.

To proceed, first handle the case where $\delta = 0$. You have to find, for each $z$, the probabilities that $F$ outputs $z$ on inputs 0 and 1. Call these probabilities $F(z|0)$ and $F(z|1)$. What linear combinations of these two variables should equal $P_X(z)$ and $P_Y(z)$ respectively? Solve for $F(z|0)$ and $F(z|1)$. What assumption allows you to be sure that the resulting numbers can be taken to be probabilities?

To handle the case where $\delta > 0$, start by proving that the probabilities of areas $A$ and $B$ are at most $\delta$. Now proceed under the assumption that both of them have area exactly $\delta$. In that case, you can write $P_X = \delta P_A + (1-\delta)P'_x$ and $P_Y = \delta P_B + (1-\delta)P'_y$, where $P_A, P_B, P'_x, andP'_y$ are probability distributions and $P'_x, P'_y$ satisfy $P'_x \approx_{(\varepsilon,0)} P'_y$. You can generate $P_A$ and $P_B$ from the inputs “I am $U$” and “I am $V$”, and use what you learned in the case $\delta = 0$ to generate $P'_x$ and $P'_y$ under appropriate distributions on 0 and 1.

Finally, extend this solution to handle the general case.

We can now proceed to the proof of Strong Composition (Theorem 1.1).

Fix a sequence of $k$ mechanisms $A_j$, each of which takes a data set in $\mathcal{X}^n$ as well as a partial transcript $y_1, \dots, y_{j-1}$ (abbreviated $\vec{y}_1^{j-1}$) such that, for every partial transcript, $A_j(\cdot; \vec{y}_1^{j-1})$ is $(\varepsilon, \delta)$-differentially private. Also, fix two data sets $\mathbf{x}, \mathbf{x}'$ that differ in one entry.

For every partial transcript $\vec{y}_1^{j-1}$, we have $A_j(\mathbf{x}; \vec{y}_1^{j-1}) \approx_{\varepsilon,\delta} A_j(\mathbf{x}'; \vec{y}_1^{j-1})$ and so there exists a randomized map $F_{\vec{y}_1^{j-1}}$ such that $F_{\vec{y}_1^{j-1}}(U)$ and $F_{\vec{y}_1^{j-1}}(V)$ have the same distributions as $A_j(\mathbf{x}; \vec{y}_1^{j-1})$ and $A_j(\mathbf{x}'; \vec{y}_1^{j-1})$, respectively.

This allows use to show the first important claim:

**Claim 3.3.** *There is a randomized map $F^*$ such that the composed mechanism $A$ satisfies:*

$$A(\mathbf{x}) \sim F^*(U_1, \dots, U_k) \text{ where } U_1, \dots, U_k \sim_{i.i.d.} U \text{ and} \tag{4}$$

$$A(\mathbf{x}') \sim F^*(V_1, \dots, V_k) \text{ where } V_1, \dots, V_k \sim_{i.i.d.} V\,. \tag{5}$$

*Proof.* Consider the algorithm:

```
Algorithm 1: $F^*(z_1, \dots, z_k)$:
for $j = 1$ to $k$ do
    $y_j \leftarrow F_{\vec{y}_1^{j-1}}(z_j)$ ;
return $(y_1, \dots, y_k)$.
```

Since $F_{\vec{y}_1^{j-1}}(U_j)$ has the same distribution as $A_j(\mathbf{x}; \vec{y}_1^{j-1})$ for each stage $j$, the overall distribution of $F^*(U_1, \dots, U_k)$ is the same as $A(\mathbf{x})$ (and similarly for $\mathbf{x}'$ when the inputs are i.i.d. copies of $V$). □

To prove that $A$ is $\tilde{\varepsilon}, \tilde{\delta}$-differentially private, it suffices, by closure under postprocessing, to prove that $(U_1, \dots, U_k) \approx_{\tilde{\varepsilon},\tilde{\delta}} (V_1, \dots, V_k)$. We are almost done!

## 3.2 Strong Composition for Leaky Randomized Response

**Claim 3.4.** *$(U_1, \dots, U_k) \approx_{\tilde{\varepsilon},\tilde{\delta}} (V_1, \dots, V_k)$ where $\tilde{\varepsilon}, \tilde{\delta}$ are as in Theorem 1.1.*

*Proof.* We'll consider two "bad events": $B_1$ and $B_2$. The first, $B_1$, is when we see a clear signal that the input was drawn according to $U$:

$$B_1 = \{\vec{z} : \text{at least one } z_j \text{ is "I am U"}\}. \tag{6}$$

If $\vec{z}$ is distributed as $U_1, \dots, U_k$, then the probability of $B_1$ is exactly $1 - (1-\delta)^k \le k\delta$.

If $\vec{z} \sim U_1, \dots, U_k$, then conditioned on $\bar{B}_{1,u}$ not occurring, we have $\vec{z} \in \{0,1\}^k$. The probability of $\vec{z}$ is nonzero under both $U$ and $V$, and we can compute the odds ratio by taking advantage of independence:

$$\ln\left(\frac{P_U(\vec{z})}{P_V(\vec{z})}\right) = \sum_j \ln\left(\frac{P_U(z_j)}{P_V(z_j)}\right) = \sum_j \ln\left(\frac{(1-\delta)e^{\varepsilon(1-z_j)}/(e^\varepsilon+1)}{(1-\delta)e^{\varepsilon(z_j)}/(e^\varepsilon+1)}\right) = \sum_j \varepsilon(-1)^{z_j}\,.$$

This log odds ratio is thus a sum of bounded, independent random variables under distribution $U$, with expectation

$$\mathop{\mathbb{E}}_{\vec{z}\sim(U_1,\dots,U_k)}\left(\frac{P_U(\vec{z})}{P_V(\vec{z})}\middle|\bar{B}_1\right) = k\varepsilon \cdot \mathbb{E}\left((-1)^U\middle|U \in \{0,1\}\right) = k\varepsilon\frac{e^\varepsilon - 1}{e^\varepsilon + 1}\,.$$

By the Chernoff bound (see appendix), for any $t > 0$ we have

$$\Pr_{\vec{z}\sim U_1,\ldots,U_k}\left(\underbrace{\ln\left(\frac{P_U(\vec{z})}{P_V(\vec{z})}\right) > \tilde{\varepsilon}}_{\text{event } B_2}\middle|\bar{B}_1\right) \le e^{-t^2/2} \text{ where } \tilde{\varepsilon} \overset{\text{def}}{=} k\varepsilon\frac{e^{\varepsilon}-1}{e^{\varepsilon}+1} + t\varepsilon\sqrt{k}.$$

Let $B_2$ be the event that $\left\{\vec{z} \in \{0,1\}^k : \ln\left(\frac{P_U(\vec{z})}{P_V(\vec{z})}\right) > k\varepsilon\frac{e^{\varepsilon}-1}{e^{\varepsilon}+1} + t\varepsilon\sqrt{k}\right\}$. Note that conditioned on $\bar{B}_1 \cap \bar{B}_2$, the ratio of $P_U(\vec{z})$ to $P_V(\vec{z})$ is bounded. Hence, for any event $E$,

$$P_U(E \cap \bar{B}_1 \cap \bar{B}_2) \le e^{\tilde{\varepsilon}} P_V(E \cap \bar{B}_1 \cap \bar{B}_2) \le e^{\tilde{\varepsilon}} P_V(E)\,.$$

This allows us to show the indistinguishability condition we want:

$$\begin{aligned} P_U(E) &\le P_U(E \cap \bar{B}_1 \cap \bar{B}_2) + P_U(B_1) + P_U(B_2|\bar{B}_1)P_U(\bar{B}_1) \\ &\le e^{\tilde{\varepsilon}} P_V(E) + k\delta + e^{-t^2/2}\,. \end{aligned}$$

Setting $t = \sqrt{2\ln(1/\delta')}$ completes the proof of Claim 3.4 and also of Theorem 1.1. □

**Exercise 3.5.** Use the proof strategy from the previous theorem to show that the composition of an $(\varepsilon_1, \delta_1)$-DP algorithm with a $(\varepsilon_2, \delta_2)$-DP algorithm is $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-DP.

### Additional Reading and Watching

The first version of the strong composition theorem appeared in [DRV10]. Our presentation is based on Kairouz et al. [KOV15], as well as Dwork and Roth [DR14, Sections 3.5.1–2]. The characterization of $(\varepsilon, \delta)$ indistinguishability of Lemma 3.1 is due to [KOV15]. Their proof is based on a much more general result of Blackwell (1953).

There are now quite a few techniques to get tighter analyses of the for the adpative composition of specific algorithms. Examples include concentrated DP [DR16, BS16, BDRS18], Renyi DP [Mir17], and Gaussian DP [DRS19]. That literature continues to evolve quickly.

**Acknowledgement** This lecture note is built on course material developed by Aaron Roth, Adam Smith, and Jonathan Ullman.

## References

[BDRS18] Mark Bun, Cynthia Dwork, Guy N Rothblum, and Thomas Steinke. Composable and versatile privacy via truncated CDP. In *Annual ACM Symposium on Theory of Computing*, STOC '18, 2018.

[BS16] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, TCC '16, 2016. https://arxiv.org/abs/1605.02065.

[DR14] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy.* NOW Publishers, 2014.

[DR16] Cynthia Dwork and Guy N Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*, 2016. https://arxiv.org/abs/1603.01887.

[DRS19] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*, 2019. https://arxiv.org/abs/1905.02383.

[DRV10] Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *FOCS*. IEEE, 2010.

[KOV15] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International Conference on Machine Learning*, ICML '15, 2015. https://arxiv.org/abs/1311.0776.

[Mir17] Ilya Mironov. Rényi differential privacy. In *IEEE Computer Security Foundations Symposium*, CSF '17, 2017. https://arxiv.org/abs/1702.07476.

# Appendix

"Chernoff bounds" are a family of concentration inequalities for sums of independent random variables. A useful example is the following:

**Lemma .6.** *Let* $X_1, \ldots, X_n$ *be i.i.d. random variables taking values in* $[0, 1]$. *Let* $X$ *denote their sum and let* $\mu = \mathbb{E}(X_i)$ *(so that* $\mathbb{E}(X) = \mu n$. *Then,*

- *For every* $\delta \geq 0$, $\mathbb{P}(X > (1+\delta)\mu n) \leq e^{-\delta^2 \mu n/3}$
- *For every* $\delta \in [0, 1]$, $\mathbb{P}(X < (1-\delta)\mu n) \leq e^{-\delta^2 \mu n/2}$.

*In particular, for every* $t > 0$, *the probability that* $|X - \mu n| \geq t\sqrt{n}$ *is at most* $2\exp(-t^2/3)$.