Tel Aviv University
Raymond and Beverly Sackler Faculty of Exact Sciences
Blavatnik School of Computer Science

Randomness Extractors and Space-Bounded Computation

A thesis submitted for the degree of
Doctor of Philosophy
by

Dean Doron

Thesis supervisor: Prof. Amnon Ta-Shma

Submitted to the Senate of Tel Aviv University
August 2018

Contents

Abstract

The role of randomness in computation is an important and fundamental one, in the computational model (can every probabilistic space- or time-bounded algorithm be derandomized?) as well as in the information-theoretic model (how can we extract true randomness from weak sources?). In this work we study problems in randomness extraction and in probabilistic space-bounded computation.

In the first part of the thesis, we address problems in extracting randomness from independent weak sources – a research area which has been extensively studied for the last three decades. Recently, there has been a burst of significant progress that lead to an explicit construction of two-source extractors supporting poly-logarithmic min-entropy [CZ16], together with a set of new primitives and techniques. The breakthrough construction of Chattopadhyay and Zuckerman used non-malleable extractors as an important ingredient. However their reduction was sub-optimal, even assuming optimal non-malleable extractors. We improve upon their result and construct two-source extractors supporting *near-logarithmic* min-entropy by giving an entropy-efficient reduction that incorporates a new sampling technique.

Although those extractors support very small min-entropies, they cannot achieve exponentially small error. We also make some advancement in the low-error regime and give two new constructions. The first construction is a conditional one, showing that good non-malleable extractors can be used to construct low-error two-source extractors supporting polynomially-small min-entropy. The second construction is an explicit one, although it only gives the weaker notion of a two-source *condenser*. The condenser has low-error, supports poly-logarithmic min-entropy and has a very small entropy gap.

Concluding the first part of the thesis, we give a new construction of almost-optimal *unbalanced* two-source extractors. These extractors yield strong, small-error, one-bit strong dispersers with near-optimal seed-length and near-optimal entropy loss. This is one of the few constructions of dispersers that outperform optimal extractors. From a coding-theoretic point of view, we construct binary erasure list-decodable codes with near-optimal list-size and near-optimal rate.

In the second part of the thesis, we study small-space computation through the linear-algebraic lens, showing that in order to derandomize small-space approximation schemes (say, of matrix inversion) it is sufficient to derandomize the corresponding decision classes. We further study the space complexity of approximating a solution to a Laplacian linear system. We give a probabilistic, logspace algorithm solving the problem even for directed graphs that mix in polynomial-time.

These results together with related work on the subject reveal a picture where the various space-bounded classes (e.g., probabilistic logspace, quantum logspace and the class DET) can be characterized by algebraic problems, where, roughly speaking, the difference between the classes lies in the kind of operators they can handle.

The work presented in this thesis relies upon [DTS15a, DLGTS17, BADTS17, BACD+18, BADTS18, BACDTS18], and is a result of joint work with Avraham Ben-Aroya, Eshan Chattopadhyay, Gil Cohen, François Le Gall, Xin Li and Amnon Ta-Shma.

Part I

Two-Source Extractors and Related Constructions

Chapter 1

Part I Overview

The problem of extracting randomness from imperfect random sources can be traced back to von Neumann [Neu51]. Ideally, and somewhat informally, a randomness extractor is an algorithm that produces, or extracts, truly random bits from an imperfect source of randomness. Going beyond that particular task, randomness extractors have found dozens of applications for error correcting codes, cryptography, combinatorics, and circuit lower bounds to name a few (see, e.g., [Sha02, Wig09] and references therein).

Ideally, a randomness extractor would have been defined as a function $\mathsf{Ext}\colon \{0,1\}^n \to \{0,1\}^m$ with the property that for every source X with sufficient entropy, the output $\mathsf{Ext}(X)$ is ε-close to the uniform distribution on $\{0,1\}^m$ in the statistical distance, which we write as $\mathsf{Ext}(X) \approx_\varepsilon U_m$.

Indeed, some *structured* sources allow for deterministic extraction. For example, such a function Ext exists for bit-fixing sources that arise naturally in cryptography [CGH+85, KZ06, GRS06, Rao09b, Gab11, CS15], affine sources ([Bou07, Yeh11, Li16] to name a few) and samplable sources [TV00, Vio14]. Unfortunately, for general imperfect sources, where we are only guaranteed that there are no heavy elements in their range, deterministic extraction is impossible.

To see the above fact, let us make things more concrete. We model a *weak source* by a random variable X that, for convenience, is assumed to be supported on n-bit strings. The by-now standard and most useful measure for the amount of randomness in X is its *min-entropy*, originally proposed by Chor and Goldreich [CG88]. The min-entropy of X is the maximum k for which one cannot guess X with probability larger than 2^{-k}, or alternatively, the maximum k for which $\Pr[X = x] \leq 2^{-k}$ for every element x in the support of X. For any such k, we say that X is an (n,k) *source*, or a k-*source* for short. Now, fix any function $\mathsf{Ext}\colon \{0,1\}^n \to \{0,1\}$, assume without loss of generality that $|\mathsf{Ext}^{-1}(0)| \geq |\mathsf{Ext}^{-1}(1)|$ and let X be uniformly distributed over $\mathsf{Ext}^{-1}(0)$. The min-entropy of X is very high, at least $n-1$, however $\mathsf{Ext}(X)$ is obviously constant.

In light of that, several types of randomness extractors, that relax in different ways the above dream definition have been introduced and studied in the literature. In *seeded extractors* we allow for additional independent and truly uniform bits. Namely, a seeded extractor $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ takes as inputs a sample $x \in \{0,1\}^n$ from an (n,k) source X and an auxiliary short seed $y \in \{0,1\}^d$ sampled uniformly at random, and outputs $\mathsf{Ext}(x,y)$. The guarantee is that $\mathsf{Ext}(X, U_d) \approx_\varepsilon U_m$. Ideally, we would like Ext to

support small k-s, have a small seed-length with a good dependence on the extraction error and have output length close to k.

In *two-source extractors* (or *multi-source extractors*) our goal is to extract from two (or more) *independent* weak sources: A function $\mathsf{2Ext}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ is a two-source extractor supporting min-entropy k is for every two independent (n,k) weak sources X and Y, it holds that $\mathsf{2Ext}(X,Y) \approx_\varepsilon U_m$. Non-explicitly, k can be as low as $\log n + 2\log(\frac{1}{\varepsilon}) + O(1)$.

In the next paragraphs we will continue discussing seeded and two-source extractors, followed by an overview of our results.

Seeded extractors. The problem of constructing explicit *seeded extractors* has been subject to prolific research over the past three decades. Originally raised in the context of fooling space-bounded computation ([NZ96] and a multitude of followup works) and simulating randomized algorithms with weak sources [Zuc96b], seeded extractors have further found a plethora of applications in theoretical computer sciences. They are also used as ingredients in constructing other important combinatorial objects, and we will see examples for that in following chapters.

Early constructions of seeded extractors were based on random walks, bounded independence and various structural transformations (e.g., [ILL89, Zuc90, NZ93, NZ96, TS96, SSZ98, SZ99b, NTS99]). Trevisan [Tre01] made a major breakthrough by establishing connections between pseudorandom generators from hard functions, error correcting codes and seeded extractors. Other than the improvement in parameters, Trevisan's extractor exhibited a new technique for the construction and analysis of extractors – the *reconstruction paradigm*, which reduces the problem of extraction to a reconstruction task from a small advice. Several constructions followed and introduced more sophisticated reconstruction extractors (e.g., [STSZ06, SU05, Uma03]).

The latter constructions achieved relatively good parameters, but it was only until the work of Lu, Reingold, Vadhan, and Wigderson [LRVW03] that extractors having logarithmic seed-length for *all* min-entropies were constructed. In 2007, Guruswami, Umans and Vadhan [GUV09] gave a direct and elegant construction of almost optimal extractors (with a better dependence on the error) based on Parvaresh-Vardy codes. See also [DW11, DKSS13, TSU12] for explicit seeded extractors that achieve better entropy loss.

Two-source extractors. Explicitly constructing good *two-source extractors* turned out to be challenging.[1] Chor and Goldreich initiated the study of two-source extractors, and proved that the inner-product function works well for min-entropy k greater than $n/2$ [CG88]. Bourgain [Bou05] gave a two-source extractor construction for $k = (1/2 - \alpha)n$ for some small constant $\alpha > 0$. Due to the difficulty of constructing good two-source extractors, another research line focused on extracting randomness from multiple sources, trying to minimize the number of sources and the min-entropy needed. This includes [BIW06, Rao09a, Li11, Li13a, Li13b, Li15b].

Eventually, two decades after Bourgain's result, Chattopadhyay and Zuckerman [CZ16] managed to drastically improve the entropy requirement and gave a two-source extractor

[1] Much work has been done trying to explicitly construct the weaker notion of two-source *dispersers*, giving rise to explicit constructions of *Ramsey graphs*. We will discuss it in Section 2.5.2.

for *poly-logarithmic* min-entropy! The breakthrough construction of [CZ16] crystalized ideas from multi-sources extractors [Rao09a, Li15b] and used *non-malleable extractors* as an important ingredient. First introduced by Dodis and Wichs [DW09] in the context of privacy amplification, non-malleable extractors strengthen the notion of seeded extractors and further guarantee that the output of the extractor is uniform even given the extractor's output on maliciously tampered seeds (see Section 2.4 for details). Several improvements on the [CZ16] construction followed shortly after, including [Mek17, Li16]. Cohen and Schulman [CS16] observed that all previous techniques for constructing multi-source extractors cannot get below min-entropy $\log^2 n$, and by introducing new techniques managed to get the first multi-source construction for near-logarithmic entropy. Chattopadhyay and Li [CL16] reduced the number of sources in such a construction to a constant and Cohen [Coh16b] put it on five.

Explicit two-source extractors for near-logarithmic min-entropy. In Chapter 3, we take a significant step towards the goal of constructing optimal two-source extractors in the high-error regime, and give the first explicit construction of two-source extractors supporting near-logarithmic min-entropy. Briefly speaking, the [CZ16] construction introduced a reduction from two-source extractors to non-malleable extractors. However, even assuming the existence of optimal explicit non-malleable extractors, their construction only gives a two-source extractor for $\mathsf{polylog}(n)$ entropy, rather than the optimal $O(\log n)$.

Our result extends the [CZ16] framework and is able to support smaller entropies by making the reduction to non-malleable extractors *entropy-efficient*. In a nutshell, the [CZ16] construction uses a seeded extractor as a *sampler*, and we observe that one could use a weaker object, a *somewhere-random condenser* having a small entropy gap, and such condensers allow for better parameters, in fact better than optimal extractors. We further constructed such condensers by employing the [RRV99] error-reduction scheme. Since our work was published, improved constructions of non-malleable extractors have emerged [Coh16d, Li17, Li18] and following our reduction we now have explicit two-source extractors supporting min-entropy as low as $O(\log n \frac{\log\log n}{\log\log\log n})$.

Low-error two-source extractors and condensers. Although recent constructions of two-source extractors come tantalizingly close to supporting $O(\log n)$ min-entropy, they all share a very unfortunate disadvantage: they do not support small extraction error.[2] Stated otherwise, their running time is not polynomial in $\log(\frac{1}{\varepsilon})$ but at best polynomial in $\frac{1}{\varepsilon}$. In Chapters 4 and 5, we make progress towards tackling the problem.

In Chapter 4, we further investigate the connection between two-source extractors and non-malleable extractors and propose a way to construct two-source extractors with *very small error*, supporting polynomially-small entropy rate, given good non-malleable extractors. The parameters we require from the non-malleable extractors for our reduction to hold fit quite comfortably in the non-explicit construction, but currently it is not known how to explicitly construct them.

The construction in Chapter 4 uses a completely different sampling technique than the low-entropy constructions use, producing a much smaller sample set with a weaker, one-

[2]We can hope for an extraction error which is exponentially-small in the input source's min-entropy.

sided, guarantee. Consequently, we are able to get rid of components that impose large error. Offhand, it is not clear whether there exist explicit samplers with the desired parameters, but remarkably, Zuckerman's dispersers ([Zuc07], see also Section 2.3.1) allow for almost optimal parameters in the regime that interests us.

In Chapter 5, we explicitly construct a relaxed notion of a two-source extractor, called a two-source condenser, in which we are only guaranteed that the output distribution will be close to a distribution having high min-entropy rather than to the uniform one. We manage to construct a two-source condenser supporting poly-logarithmic min-entropy and running in time $\mathsf{poly}(n, \log \frac{1}{\varepsilon})$. We do this by extending the notion of *resilient functions*, where high error is unavoidable, to *entropy-resilient functions*, that output many bits (much more than in previous works) and allow for low error. The two-source condensers we obtain also achieve a very small *entropy gap*, namely, the min-entropy in the m-bits output distribution is as large as $m - o(\log \frac{1}{\varepsilon})$.

Unbalanced almost-optimal two-source extractors and related constructions. In Chapter 6, we construct new, almost-optimal two-source extractors in the unbalanced regime (i.e., where the sources are not of the same length). Our extractors extract one bit with constant error from a source of length n with min-entropy $O(\log \log n)$ and an independent source of length $O(\log n)$ with arbitrarily small constant min-entropy rate.

We show that these extractors give rise to almost optimal one-bit *strong dispersers*, which will be defined later on, with near-optimal seed-length and near-optimal entropy loss. Equivalently, our construction also gives near-optimal binary *erasure list-decodable codes*. The codes we construct can be list-decoded from $1 - \varepsilon$ fraction of adversarial erasures, with near-optimal list-size of $\mathsf{polylog}(1/\varepsilon)$ and near-optimal rate $O(\varepsilon^{1+\delta})$, where δ is an arbitrarily small constant. This is the first construction to break the rate $O(\varepsilon^2)$ barrier, solving a longstanding open problem from [Gur04b, GI02, Gur04a]. The constructions of Chapter 6 use a combination of recent pseudorandomness machinery, together with a delicate and novel analysis needed in order to solve dependence and error issues.

As often happens in this area of research, there is a large interplay between various extractors and between extractors and other pseudorandomness objects, and the results in this thesis are of no exception. Before delving into the results, in Chapter 2 we formally present some of the primitives we will use and give some preliminary results.

Chapter 3 follows the results in [BADTS17], Chapter 4 follows [BACD+18], Chapter 5 follows the construction in [BACDTS18] and Chapter 6 is due to [BADTS18].

Chapter 2

A Myriad of Pseudorandom Primitives

In theoretical computer science, we say a fixed object satisfying a desirable property is *pseudorandom* if it "random-like", i.e., a random function also has that desirable property with high probability. Alternatively, we say it is pseudorandom if a certain desirable family of tests cannot distinguish its output from that of a truly random object.

Notable examples can be found in extractor theory, which is the focal point of this part of the thesis.[1] In Chapter 1 we talked about seeded extractors and two-source extractors. In this section we give a more formal treatment of randomness extractors, as well as present some additional pseudorandom primitives. We will use all of these primitives (and more) for our results in the following chapters.

2.1 Preliminary Definitions

Throughout, we use the convention that lowercase variables are the logarithm (in base-2) of their corresponding uppercase variables, e.g., $n = \log N$, $d = \log D$, $a = \log A$, $r = \log R$, $r' = \log R'$, etc. The density of a set $B \subseteq [D]$ is $\rho(B) = \frac{|B|}{D}$. We denote by $[A]$ the set $\{1, \ldots, A\}$.

We say a function $f\colon A \to B$ is *explicit* if there exists a deterministic polynomial algorithm that runs in time $\mathsf{polylog}(|A|)$ and computes f. For a function $f\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ and sets $A \subseteq \{0,1\}^n$ and $B \subseteq \{0,1\}^d$ we denote $f(A,B) = \{f(a,b) : a \in A \wedge b \in B\}$.

2.1.1 Random variables and min-entropy

Definition 2.1.1. *The* statistical distance *between two random variables X and Y on the same finite domain Ω is defined as*

$$|X - Y| = \frac{1}{2}\sum_{a\in\Omega} |\Pr[X = a] - \Pr[Y = a]| = \max_{A\subseteq\Omega}(\Pr[X \in A] - \Pr[Y \in A]).$$

[1] Other prominent and perhaps more well-known examples are expander graphs (e.g., with respect to the property of mixing) and error-correcting codes (e.g., with respect to list-decoding capabilities).

If $|X - Y| \leq \varepsilon$ we say X is ε-close to Y and denote it by $X \approx_\varepsilon Y$. ◊

We will denote by U_n the random variable distributed uniformly over $\{0,1\}^n$. We say a random variable is *flat* if it is uniform over its support. For a function $f\colon \Omega_1 \to \Omega_2$ and a random variable X supported on Ω_1, $f(X)$ is the random variable supported on Ω_2 obtained by choosing x according to X and computing $f(x)$. For every $f\colon \Omega_1 \to \Omega_2$ and two random variables X and Y supported on Ω_1, it holds that $|f(X) - f(Y)| \leq |X - Y|$. We will make use of the following lemma.

Lemma 2.1.2. *Let $X_1, \ldots, X_t, Y_1, \ldots, Y_k$ be random supported on $\{0,1\}^m$. Further suppose that for any $i \in [t]$,*

$$\left(X_i, \{X_j\}_{j \neq i}, Y_1, \ldots, Y_k\right) \approx_\varepsilon \left(U_m, \{X_j\}_{j \neq i}, Y_1, \ldots, Y_k\right).$$

Then, $(X_1, \ldots, X_t, Y_1, \ldots, Y_k) \approx_{t\varepsilon} (U_{tm}, Y_1, \ldots, Y_k)$.

Definition 2.1.3. *The* min-entropy *of a random variable X is defined by*

$$H_\infty(X) = \min_{x \in \mathrm{Supp}(X)} \log \frac{1}{\Pr[X = x]}.$$

A random variable X is an (n,k) source if X is supported on $\{0,1\}^n$ and has min-entropy at least k. When n is clear from the context we sometimes omit it and simply say that X is a k-source. ◊

Every k-source X can be expressed as a convex combination of *flat* distributions, each with min-entropy at least k (see, e.g., [Vad12, Lemma 6.10]).

Definition 2.1.4. *Let X, Y be two random variables. The* average conditional min-entropy *of X given Y is*

$$\tilde{H}_\infty(X|Y) = -\log\left(\mathbb{E}_{y \sim Y}\left[2^{-H_\infty(X|Y=y)}\right]\right).$$

◊

We will use the following simple claim about average conditional min-entropy.

Claim 2.1.5. *For any random variables X, Y,*

$$\tilde{H}_\infty(X|Y) \geq H_\infty(X) - \log|\mathrm{Supp}(Y)|.$$

Definition 2.1.6. *For $\varepsilon \geq 0$, the* smooth min-entropy $H^\varepsilon_\infty(X)$ *is the supremum of $H_\infty(X')$ over all distributions $X' \approx_\varepsilon X$.* ◊

We will use the following easy claim.

Claim 2.1.7. *If $H^{1/2}_\infty(X) \geq k$ then $|\mathrm{Supp}(X)| \geq 2^{k-1}$.*

2.1.2 Somewhere-random sources

We now define weak source that have an additional structure. Roughly, a somewhere-random source is partitioned into blocks, and at least one block contains the desirable min-entropy. Formally:

Definition 2.1.8. *A source $X = X_1 \circ \ldots \circ X_A$ is an $(n, k, (\alpha, \beta))$* somewhere-random (s.r.) source *if there exists a random variable $I \in \{0, \ldots, A\}$ such that for every $i \in [A]$, $H^{\alpha}_{\infty}(X_i | I = i) \geq k$ and $\Pr[I = 0] \leq \beta$. The variable I is called the* indicator *of the source. If $\alpha = \beta = 0$ we say X is a k* s.r. source. *We say X is a (n, k, ζ)* s.r. source *if X is ζ-close to a k s.r. source over $\{0,1\}^n$.* ◊

Claim 2.1.9. *Let X be an $(n, k, (\alpha, \beta))$ s.r. source. Then, X is a $(n, k, \alpha + \beta)$ s.r. source.*

Intuitively, it is often convenient to think of a k s.r. source $X = X_1 \circ \ldots \circ X_A$ as if one of the blocks X_i is having k min-entropy, and the other blocks are arbitrarily correlated with it. Formally, X is a k s.r. source if it is a convex combination of such sources.

2.1.3 Limited independence and non-oblivious bit-fixing sources

Definition 2.1.10. *A distribution X over $\{0,1\}^n$ is called* (t, δ)-wise independent *if the restriction of X to every t coordinates is γ-close to U_t. For $\delta = 0$ this is simply the notion of a t-wise independent distribution.* ◊

Every (t, δ)-wise independent is close to some t-wise independent distribution.

Lemma 2.1.11 ([AGM03])**.** *Let $X = X_1, \ldots, X_n$ be a distribution over $\{0,1\}^n$ that is (t, δ)-wise independent. Then, X is $(n^t \delta)$-close to a t-wise independent distribution.*

Next, we define non-oblivious bit-fixing sources, followed by resilient functions.

Definition 2.1.12. *A source X over $\{0,1\}^A$ is called a (q, t, δ)* non-oblivious bit-fixing source *if there exists a subset $Q \subseteq A$ of size at most q such that the joint distribution of the bits in $A \setminus Q$ is (t, δ)-wise independent. The bits in Q are allowed to arbitrarily depend on the bits in $A \setminus Q$. If $\delta = 0$ we often say that X is a (q, t) non-oblivious bit-fixing source.* ◊

A q-resilient function $f \colon \{0,1\}^A \to \{0,1\}$ can be thought of as an A-players game. If all players feed uniform and independent inputs to f, the output distribution has small bias, and, furthermore, this property is retained even if any q players decide to deviate from the rules of the game and choose their inputs as a function of all other inputs to f. Formally:

Definition 2.1.13. *Let $f \colon \{0,1\}^A \to \{0,1\}$, $\mathcal{D}$ a distribution over $\{0,1\}^A$ and $Q \subseteq A$. Let $I_{Q,\mathcal{D}}(f)$ denote the probability that f is undetermined when the variables outside Q are sampled from $\mathcal{D}$. We define $I_{q,t,\delta}(f)$ to be the maximum of $I_{Q,\mathcal{D}}(f)$ over all $Q \subseteq A$ of size q and all $\mathcal{D}$ that is a (t, δ) independent distribution.*

We say that f is (t, δ) independent (q, ε) resilient *if $I_{q,t,\delta}(f) \leq \varepsilon$. When we say a function is (q, t, ε) resilient we mean that it is $(t, 0)$ independent (q, ε) resilient, and if we omit the ε we mean that it is (q, t, ε) resilient for some non-trivial ε.* ◊

Balanced resilient functions can be seen as deterministic one-bit extractors against non-oblivious bit-fixing sources and are also analogous to one-round collective coin flipping protocols [BOL85].

The work of Viola [Vio14] shows that for every $\alpha > 0$, the majority function over n bits is $(t, 0)$ independent $(n^{1/2-\alpha}, O(\frac{\log t}{t} + n^{-\alpha}))$ resilient. Combining this with 2.1.11, we conclude:

Lemma 2.1.14. *There exists a constant c_{maj} such that for every $\alpha > 0$ and a $(q = n^{\frac{1}{2}-\alpha}, t, \delta)$ non-oblivious bit-fixing source X on n bits,*

$$\left|\Pr[\mathsf{maj}(X_1, \ldots, X_n) = 1] - \frac{1}{2}\right| \leq c_{\mathsf{maj}} \cdot \left(\frac{\log t}{t} + n^{-\alpha} + \delta n^t\right).$$

Chattopadhyay and Zuckerman [CZ16] derandomized the Ajtai-Linial function [AL93] and constructed a (monotone) resilient function that handles $q = n^{1-\alpha}$ for *any* constant α. Their construction was later improved in [Mek17].

Theorem 2.1.15 ([CZ16, Mek17])**.** *For every $0 < \gamma < 1$ there exists a constant $c_\gamma \geq 1$ such that for every integer n there exists an explicit function $f\colon \{0,1\}^n \to \{0,1\}$ with the following property. For every $t \geq c_\gamma \log^4 n$,*

- *f is almost balanced: For any t-wise independent distribution $\mathcal{D}$ on $\{0,1\}^n$,*

$$\Pr_{x\sim\mathcal{D}}[f(x) = 1] = 1/2 \pm n^{-1/c_\gamma}, \text{ and,}$$

- *f is resilient: $I_{q,t,0}(f) \leq c_\gamma \cdot \frac{q}{n^{1-\gamma}}$.*

2.2 Seeded Extractors

We start with the influential notion of seeded extractors.

Definition 2.2.1 ([NZ96])**.** *A function $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k,ε)* extractor *if for every (n,k) source X and for Y that is uniform over $\{0,1\}^d$ and independent of X, it holds that $\mathsf{Ext}(X,Y) \approx_\varepsilon U_m$. We say that Ext is* strong *if $(\mathsf{Ext}(X,Y), Y) \approx_\varepsilon U_m \times Y$.* ◊

A strong extractor outputs a string coming from a distribution which is not only close to uniform, but whose randomness is (almost) independent of the seed Y. This property has many applications, one of which is in *privacy amplification* protocols, in which one party sends Y over a public channel [BBR88].

Non-explicitly, very good strong extractors exist, and their parameters are matched by a corresponding lower bound.

Theorem 2.2.2 ([RTS00])**.** *For all integers n, k and every $\varepsilon > 0$ there exists a strong (k,ε) extractor $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $m = k - 2\log(\frac{1}{\varepsilon}) - O(1)$ and $d = \log(n-k) + 2\log(\frac{1}{\varepsilon}) + O(1)$.*

Conversely, if $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a strong (n,k) extractor then $k \geq m + 2\log(\frac{1}{\varepsilon}) - O(1)$ and $d \geq \log(n-k) + 2\log(\frac{1}{\varepsilon}) - O(1)$.

Explicit constructions of seeded extractor optimal up to constant factors exist.

Theorem 2.2.3 ([GUV09])**.** *There exists a constant $c_{\mathsf{GUV}} > 0$ such that the following holds. For all integers n, k and every $\varepsilon > 0$ there exists an explicit strong (k,ε) extractor $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ having seed length $d = c_{\mathsf{GUV}} \log(\frac{n}{\varepsilon})$ and $m = \frac{k}{2}$ output bits.*

If we want to output almost all the entropy, instead of only a constant fraction of it, we can do it explicitly at the cost of a longer seed.

Theorem 2.2.4 ([GUV09])**.** *There exists a constant $c_{\mathsf{GUV}} > 0$ such that the following holds. For all integers n, k and every $\varepsilon > 0$ such that $k \geq 2\log(1/\varepsilon) + O(1)$, there exists an explicit strong (k, ε) extractor $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ having seed-length $d = c_{\mathsf{GUV}} \log n \cdot \log \frac{n}{\varepsilon}$ and $m = k - 2\log\frac{1}{\varepsilon} - O(1)$ output bits.*

Both extractors run in time which is polynomial in n and $\log(\frac{1}{\varepsilon})$.

2.2.1 Seeded extractors and samplers

Seeded extractors are tightly related to the well-known and important problem of *sampling*. Say that we want to estimate the density of some subset inside a large domain. More generally, given oracle access to a function $f\colon \{0,1\}^m \to [0,1]$, say that we want to estimate $\mu(f) = \mathbb{E}[f(U_m)]$ to within an additive error of ε.

It is easy to see that deterministically we must make at least $\Omega(2^m)$ queries to f for a reasonable ε. Using randomness, we can decrease this number drastically. By choosing $x_1, \ldots, x_T \in \{0,1\}^m$ uniformly at random for $T = O\left(\frac{\log(1/\delta)}{\varepsilon^2}\right)$ and outputting $\frac{1}{T}\sum_{i=1}^{T} f(x_i)$, we are guaranteed by the Chernoff bound that with probability at least $1-\delta$, the output deviates from $\mu(f)$ by at most ε. Note that T is independent of m.

Can we sample efficiently using less randomness? Can we do this using weak sources? Let us concentrate on *density* samplers (i.e., where the function f is the characteristic function of some set) and define things formally.

Definition 2.2.5. *Let $S\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$.*

- *We say $x \in \{0,1\}^n$ is ε-*bad for $B \subseteq \{0,1\}^m$ *if*
$$\left| \Pr_{y \sim U_d}[S(x,y) \in B] - \rho(B) \right| > \varepsilon.$$
- *We say S is a (δ, ε) sampler if for every $B \subseteq \{0,1\}^m$ we have that*
$$|\{x \in \{0,1\}^n : x \text{ is } \varepsilon\text{-bad for } B\}| < \delta N.$$

$\Diamond$

That is, we can view S as a bipartite graph, and S is a sampler if for each subset B on the right hand side, most vertices on the left hand side estimate $\rho(B)$ well, in the sense that the fraction of neighbors that fall into B is roughly $\rho(B)D$. In that setting of parameters, n is therefore the number of random bits required for sampling.

Now, if we only have a k-source at our disposal, and δ is small enough compared to k, we can still use S for sampling. Namely:

Claim 2.2.6. *Let $S\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a (δ, ε) sampler and let X be an (n,k) source. Then, for every $B \subseteq \{0,1\}^m$, we have that*
$$\Pr_{x \sim X}\left[\left||\Gamma(x) \cap B| - \rho(B)D\right| \leq \varepsilon D\right] \geq 1 - \delta \cdot \frac{N}{K},$$
where $\Gamma(x)$ denotes the neighbors of x in S.

As promised, seeded extractors and samplers are tightly related, and in fact are equivalent, at least up to some loss in parameters.

Claim 2.2.7. *Let* $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a* (k,ε) *extractor. Then,* Ext *is also a* $(\delta = \frac{K}{N}, \varepsilon)$ *sampler.*

Proof. Fix $B \subseteq \{0,1\}^m$ and let X be the set of x-s which are ε-bad for B. Assume towards a contradiction that $|X| \geq K$ and identify X with the flat distribution over its support, so $H_\infty(X) \geq k$. By the extractor property, $|\mathsf{Ext}(X, U_d) - U_m| \leq \varepsilon$, so specifically

$$|\Pr[\mathsf{Ext}(X, U_d) \in B] - \Pr[U_m \in B]| \leq \varepsilon,$$

and $\Pr[U_m \in B] = \rho(B)$. As every $x \in X$ is ε-bad for B, $|\Pr[\mathsf{Ext}(X, U_d) \in B] - \rho(B)| > \varepsilon$, in contradiction. To conclude, note that $\delta = \rho(X) = \frac{K}{N}$. □

Claim 2.2.8. *Let* $S\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a* (δ,ε) *sampler. Then,* S *is also a* $(k, 2\varepsilon)$ *extractor for* $k = n - \log(\frac{1}{\delta}) + \log(\frac{1}{\varepsilon})$.

Proof. Let X be a flat (n,k) source (the claim for general weak source follows from convexity). For every test $T\colon \{0,1\}^m \to \{0,1\}$, define Bad_T as the set of x-s which are ε-bad for T. As S is a (δ,ε) sampler,

$$\Pr_{x \sim X}[x \in \mathsf{Bad}_T] \leq \frac{\delta N}{K} = \varepsilon.$$

Now,

$$\Pr[T(S(X, U_d)) = 1] \leq \Pr[X \in \mathsf{Bad}_T] + \Pr[T(S(X, U_d)) = 1 | X \notin \mathsf{Bad}_T].$$

By the sampler property, for every $x \notin \mathsf{Bad}_T$ it holds that $\Pr[T(S(x, U_d)) = 1] \in [\rho(T) - \varepsilon, \rho(T) + \varepsilon]$, so $\Pr[T(S(X, U_d)) = 1] \leq \rho(T) + 2\varepsilon$. Also,

$$\begin{aligned}\Pr[T(S(X, U_d)) = 1] &\geq \Pr[T(S(X, U_d)) = 1 | X \notin \mathsf{Bad}_T] \Pr[X \notin \mathsf{Bad}_T] \\ &\geq (\rho(T) - \varepsilon)(1 - \varepsilon) \geq \rho(T) - 2\varepsilon.\end{aligned}$$

Overall, $|\Pr[T(S(X, U_d)) = 1] - \Pr[T(U_m) = 1]| \leq 2\varepsilon$. As this is true for every T, $|S(X, U_d) - U_m| \leq 2\varepsilon$, and we are finished. □

Following Claim 2.2.6 and Claim 2.2.7, we can summarize the sampling property that extractors induce.

Theorem 2.2.9 ([Zuc97])**.** *Let* $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a* (k_1,ε) *extractor. Identify* $\{0,1\}^d$ *with* $[D]$ *and define* $\mathrm{Samp}(x) = \{\mathsf{Ext}(x,1), \ldots, \mathsf{Ext}(x,D)\}$.

Let X *be an* (n, k_2) *source. Then, for every* $B \subseteq \{0,1\}^m$*, we have that*

$$\Pr_{x \sim X}\left[\left|\frac{|\mathrm{Samp}(x) \cap B|}{D} - \rho(B)\right| \leq \varepsilon\right] \geq 1 - 2^{k_1 - k_2}.$$

For a more elaborate discussion about samplers, including explicit constructions, refer to [Gol11b]. The viewpoint of extractors (and relaxation of extractors) as samplers will be very helpful in upcoming chapters.

2.3 Dispersers, Condensers and Mergers

In what follows, we will introduce primitives that are all relaxations of seeded extractors.

1. In a *disperser*, we require the output distribution to *cover* all but ε-fraction of its codomain and do not insist on doing so uniformly.

2. In a *condenser*, we require the output distribution to be close to having large min-entropy and not necessarily close to having full min-entropy. A good condenser increases the entropy-rate, which is the ratio between the min-entropy and the length, thus making the source more *condensed*.

3. In a *merger*, we assume our weak source has an additional structure where roughly speaking, the entropy lies in consecutive bits. That is, our source is partitioned into blocks and the guarantee is that one block has high min-entropy. To goal is to output only one block which is close to having very high min-entropy.

By setting the bar lower, it is possible to obtain explicit constructions for the above primitives that outperform optimal seeded extractors (say, getting a seed-length smaller than $2\log(\frac{1}{\varepsilon})$, or which is independent of n). Although those objects were sometimes used as an intermediate step towards constructing seeded extractors, they also found independent applications in theoretical computer science.

2.3.1 Seeded dispersers

We start with the definition of a seeded disperser.

Definition 2.3.1. *A function* $\Gamma\colon [N]\times[D]\to[M]$ *is a* (K,K') disperser *if for every* $A\subseteq[N]$ *with* $|A|\geq K$*, it holds that* $|\Gamma(A,[D])|>K'$. $\Diamond$

If $K'=(1-\varepsilon)M$ then for every (n,k) source X, the support of $\Gamma(X,U_d)$ has cardinality at least $(1-\varepsilon)M$. This is indeed a weaker than an extractor, which would have required $\Gamma(X,U_d)$ to be ε-close to U_m. In particular, in dispersers, we allow the output distribution to have heavy elements.

One can also take the samplers view of extractors, and compare them to dispersers. In a sampler, we require a large fraction of the vertices on the left hand side to estimate the *density* of any set on the right hand side to within an additive accuracy. With a disperser, we are only guaranteed that a large fraction of the vertices on the left hand side will *hit* every large enough set. One can readily see that with extractors we can simulate two-sided error probabilistic algorithms using weak sources, whereas dispersers allow for one-sided error simulation. Indeed, that was one of the early motivations for studying dispersers and later other applications emerged (see, e.g., [CW89, Sip88, Zuc96a, SSZ98, SZ99b, TS02]).

In times where explicit, almost optimal extractors exist, disperser are interesting when they beat optimal extractors, and indeed they can.

Theorem 2.3.2 ([RTS00])**.** *For all integers* n,k*, and every* $\varepsilon>0$ *there exists a* $(K,(1-\varepsilon)M)$ *disperser* $\Gamma\colon \{0,1\}^n\times\{0,1\}^d\to\{0,1\}^m$ *with* $m=k+d-\log\log(\frac{1}{\varepsilon})-O(1)$ *and* $d=\log(n-k)+\log(\frac{1}{\varepsilon})+O(1)$.

Conversely, if $\Gamma\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* $(K,(1-\varepsilon)M)$ *disperser then* $k+d \geq m+\log\log(\frac{1}{\varepsilon})-O(1)$ *and* $d \geq \log(n-k)+\log(\frac{1}{\varepsilon})-O(1)$.

We will talk more about the parameters of optimal dispersers in Chapter 6, where we will also introduce *strong* dispersers.

It is also interesting to consider the very high error regime, where $K' \ll M$. This setting of parameters is useful when we are interesting in avoiding small sets (and we will).

Theorem 2.3.3 ([RTS00])**.** *There exists a constant* c_0 *such that the following holds. Let* $\Gamma\colon [N]\times[D] \to [M]$ *be a* (K,K') *disperser where* $K < N$ *and* $K' < M/2$*. Then,* $D \geq c_0 \cdot \frac{\log \frac{N}{K}}{\log \frac{M}{K'}}$.

In what follows, we describe a beautiful construction due to Zuckerman [Zuc07] in the high error regime, achieving a *constant* degree disperser in a very elegant way.

2.3.1.1 A constant-degree disperser from additive combinatorics

Theorem 2.3.4 ([Zuc07])**.** *There exists a constant* c_{Disp} *such that the following holds. For all constants* $0 < a, b < 1$*, all integers* N*,* $K = N^a$*,* $M \leq K^{1-b}$ *and* $K' < M$ *there exists an explicit family of* (K,K') *dispersers*

$$\Gamma\colon [N] \times [D] \to [M]$$

with degree $D = c_{\mathsf{Disp}} \cdot \frac{\log \frac{N}{K}}{\log \frac{M}{K'}} = O\left(\frac{n}{\log \frac{M}{K'}}\right)$.

Note that the parameters of Theorem 2.3.4 match the lower bound given in Theorem 2.3.3 up to constant factors.

At the heart of Zuckerman's construction is the following degree-two disperser.

Lemma 2.3.5. *For every constant* $\gamma > 0$ *there exists a small constant* $\alpha > 0$ *such that the following holds. Let* p *be a prime number,* $q = 2^p$ *and set* $N = q^3$ *and* $M = q^2$*. For every* $K \leq M^{1-\gamma}$ *there exists an explicit function*

$$\Gamma\colon [N] \times [2] \to [M]$$

that is a $(K, K' = K^{\frac{2}{3}+\frac{4}{9}\alpha})$ *disperser.*

As this disperser is both simple and elegant we will give its analysis here.

Proof. Let $V = \mathbb{F}_q^2$ be the set of points over $\mathbb{F}_q$ and let $W = \mathbb{F}_q^2$ be the set of lines over $\mathbb{F}_q$. Let E be all pairs $(p,\ell) \in V \times W$ for which p and ℓ are incident. Given $e = (p,\ell) \in E$, set $\Gamma(e,1) = p$ and $\Gamma(e,2) = \ell$.

The correctness relies on the deep theorem about point-line incidences by Bourgain, Katz and Tao [BKT04].

Theorem 2.3.6 ([BKT04, BGK06])**.** *Let* q *be either a prime number or* 2^p *for some prime number* p*. Let* P, L *be sets of points and lines in* $\mathbb{F}_q^2$ *of cardinality at most* $R \leq q^{2-\gamma}$ *for some constant* $\gamma > 0$*. Then, there exists a constant* $\alpha = \alpha(\gamma) > 0$ *such that*

$$I(P,L) \;=\; O(R^{3/2-\alpha})$$

where $I(P, L)$ is the number of incidences, i.e., the number of ordered pairs (p, ℓ) for which p lies on ℓ.[2]

Now, let $A \subseteq [N]$ be a set of cardinality at least K and assume towards a contradiction that there exists a set $B \subseteq [M]$ of size at most K' for which $\Gamma(A, [2]) \subseteq B$. On the one hand, $K' \leq M^{1-\gamma}$ so $I(B, B) \leq c \cdot (K')^{3/2-\alpha}$ for some positive constants c and $\alpha = \alpha(\gamma)$ that is guaranteed by Theorem 2.3.6. On the other hand, $I(B, B) = |A| \geq K$, so overall $K' \geq c' K^{\frac{1}{3/2-\alpha}} > K^{\frac{2}{3}+\frac{4}{9}\alpha}$ for some other constant c' and a large enough K, in contradiction. □

Observe that $\frac{k'}{m} = (1 + \frac{2}{3}\alpha)\frac{k}{n} > \frac{k}{n}$, as $\alpha > 0$. For $\alpha = 0$, a more straightforward argument can be used instead of Theorem 2.3.6. However, the fact that $\frac{k'}{m} > \frac{k}{n}$ will be crucial for us when we talk about condensers, and in fact state that Γ satisfies an even stronger guarantee.

In the next chapter we will use the following property of Zuckerman's disperser, which readily follows from the construction in [Zuc07].

Claim 2.3.7. *Let $\Gamma\colon [N] \times [D] \to [M]$ be the disperser of Theorem 2.3.4. Then, for every $i \in [D]$ we have that $\Gamma(U_n, i) = U_m$.*

2.3.2 Seeded condensers

The *entropy rate* of an (n, k) source is $\frac{n}{k}$ (so the uniform distribution has entropy rate 1). In a condenser, we are interested in increasing the entropy rate, with the help of an auxiliary uniform seed.

Definition 2.3.8. *A function $\mathsf{Cond}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a $(k \to k', \varepsilon)$ condenser, if for every (n, k) source X, $\mathsf{Cond}(X, U_d)$ is ε-close to an (m, k') source. If $k = \delta n$ and $k' = \delta' m$ we say Cond is a rate $(\delta \to \delta', \varepsilon)$ condenser.* ◊

Condensers are natural stepping stones towards constructing extractors (few examples are [RSW00, LRVW03, TSUZ07, GUV09]), and have also found other independent applications (e.g., [BMRV02, DPW14, CI17]). Later on, we will see how to use condensers for sampling in a certain range of parameters.

The parameters of condensers can outperform optimal extractors. For example, they can be *lossless*. We say that a condenser is lossless if it preserves the min-entropy in the system, namely if $k' = k + d$. This is in contrast to an inevitable entropy loss for extractors. Viewing lossless condensers as unbalanced bipartite graphs, they have a very strong expansion property.

Theorem 2.3.9 ([TSUZ07])**.** *$C\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a $(k \to k' = k + d, \varepsilon)$ condenser if and only for every set $A \subseteq [N]$ of size K, $|C(A, [D])| \geq (1 - \varepsilon)DK$.*

Also, the dependence of the seed-length on ε can be better than $2\log(\frac{1}{\varepsilon})$ and in fact such explicit constructions exist. The following beautiful result, based on list-decoding of Parvaresh-Vardy codes, was given by Guruswami, Umans and Vadhan.

Theorem 2.3.10 ([GUV09])**.** *For every constant $0 < \alpha < 1$, all integers n, k and every $\varepsilon > 0$ there exists an explicit $(k \to k + d, \varepsilon)$ condenser $\mathsf{Cond}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = (1 + 1/\alpha)(\log n + \log k + \log(1/\varepsilon)) + O(1)$ and $m \leq 2d + (1 + \alpha)k$.*

[2] In a certain range of parameters, the constant α was successfully brought up to $\frac{1}{6}$, which is an upper bound [Gro14].

2.3.3 Somewhere-random condensers

For seeded condensers we needed an independent uniform random string, and we were guaranteed that with high probability the output distribution had high min-entropy. In a somewhere-random condenser, the guarantee is weaker: Only *some* output block is guaranteed to have high min-entropy. A somewhere-random condenser can be seeded or seedless.

Definition 2.3.11. *A function* $\mathsf{SRC}\colon [N] \times [D] \times [A] \to [M]$ *is a* $(k \to k', \zeta)$ s.r. condenser *if for every* (n, k) *source* X *it holds that*

$$\mathsf{SRC}(X, U_d) = \mathsf{SRC}(X, U_d, 1) \circ \ldots \circ \mathsf{SRC}(X, U_d, A)$$

is an (m, k', ζ) *s.r. source.* D *is the* degree *of the s.r. condenser, and* A *is its number of blocks. If* $D = 1$ *(i.e.,* $d = 0$*) we say* SRC *is* seedless*. A condenser is a s.r. condenser with just one block.* ◊

A s.r. condenser implies a disperser with a large error (that is, in the regime where $K' \ll M$). Concretely,

Lemma 2.3.12. *Suppose* $\mathsf{SRC}\colon [N] \times [D] \times [A] \to [M]$ *is a* $(k \to k', \zeta = \frac{1}{2})$ *s.r. condenser. Define* $\Gamma\colon [N] \times [D \cdot A] \to [M]$ *by* $\Gamma(x; (y, a)) = \mathsf{SRC}(x, y, a)$*. Then,* Γ *is a* $(K, \frac{K'}{2})$ *disperser.*

Proof. Let $B \subseteq [N]$ be an arbitrary set such that $|B| \geq K = 2^k$. Then,

$$\mathsf{SRC}(B, U_d) = \mathsf{SRC}(B, U_d, 1) \circ \ldots \circ \mathsf{SRC}(B, U_d, A)$$

is $\zeta = 1/2$-close to a k' s.r. source, with some indicator random variable I. Pick any index $i \neq 0$ in the support of I. Then, conditioned on $I = i$, we have that $\mathsf{SRC}(B, U_d, i)$ is $1/2$-close to a k'-source. Thus, by Claim 2.1.7, conditioned on $I = i$, $\mathsf{SRC}(B, U_d, i)$ covers at least $2^{k'-1}$ vertices from $[M]$. But then, even without the conditioning, $\mathsf{SRC}(B, U_d, i)$ covers at least $2^{k'-1} = K'/2$ vertices from $[M]$. □

Recalling Zuckerman's degree-two disperser from Theorem 2.3.4, a more careful analysis in fact shows the same function is a seedless s.r. condenser.

Theorem 2.3.13 ([Zuc07])**.** *For every constant* $\gamma < 0$ *there exists a small constant* $\alpha > 0$ *such that the following holds. Let* p *be a prime number,* $q = 2^p$ *and set* $N = q^3$ *and* $M = q^2$*. For every* $k \leq (1 - \gamma)m$ *there exists an explicit function*

$$\mathsf{SRC}\colon [N] \times [1] \times [A = 2] \to [M]$$

that is a rate $(\delta \to (1 + \frac{\alpha}{2})\delta, N^{-\Omega(1)})$ *s.r. seedless condenser with two blocks, where* $\delta = \frac{k}{n}$*.*

We note that a preceding construction of a seedless s.r. condenser outputting a constant number of block, also using additive combinatorics, was given by Barak et al. [BKS+10].

In Chapter 3 we will construct a new, *seeded*, s.r. condenser.

2.3.4 Mergers

In a merger, we assume our weak source of randomness has the special structure of a s.r. source. That is, we take as input a list of possibly correlated random variables along with a short uniform seed and output one random variable which is close to having high min-entropy, provided at least one of the input variables has high min-entropy.

Definition 2.3.14. *A function* $\mathsf{Merg}\colon (\{0,1\}^n)^L \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k, k', ε) *merger, if for every* (n,k) *s.r. source* $X = X_1 \circ \ldots \circ X_L$*, the output* $\mathsf{Merg}(X, U_d)$ *is* ε*-close to a* k'*-source.* ◊

Mergers were first introduced by Ta-Shma [TS96] and similar to condensers, they were used for constructing extractors.

A notable merger is the *curve merger*, given by Dvir and Wigderson [DW11] and is computed as follows. Let q be a prime power. We are given a s.r. source $X = (X_1, \ldots, X_L)$ and an independent uniform seed Y. Identify $y \sim Y$ with an element of $\mathbb{F}_q$ and each $x_i \sim X_i$ with an element of $\mathbb{F}_q^n$. The merger computes the canonical curve that passes through $x_1, \ldots, x_L$ and outputs the y-th point on that curve.

In their analysis, Dvir and Wigderson used the polynomial method, which is widely applied in list-decoding algorithms as well as in the analysis of the [GUV09] condenser. The analysis was later improved by Dvir et al. [DKSS13] using the method of multiplicities, giving the following result.

Theorem 2.3.15 ([DKSS13])**.** *There exists a constant* $c_{\mathsf{DKSS}} \geq 1$ *such that the following holds. Fix* $\beta, \delta, \varepsilon > 0$*. There exists an explicit function* $\mathsf{Merg}\colon (\{0,1\}^n)^L \times \{0,1\}^d \to \{0,1\}^n$ *that is a* $(k = \delta n, k' = (1-\beta)\delta n, \varepsilon)$ *merger, with* $d = c_{\mathsf{DKSS}} \cdot \frac{1}{\beta} \log \frac{L}{\varepsilon}$*.*

Notice that d only depends on the number of blocks L and the desired error ε, but *not* on the block length n, and this remarkable property will turn crucial for us.

2.4 Non-Malleable Extractors

Non-malleable extractors, first studied by Dodis and Wichs [DW09], *strengthen* the notion of strong extractors. In a strong extractor, the output of the extractor is guaranteed to be uniform even conditioned on the seed. In a non-malleable extractor, the output of the extractor is guaranteed to be uniform conditioned on the seed and moreover on the output of the extractor applied with several adversarially-tampered seeds.

Definition 2.4.1. *A function* $\mathsf{nmExt}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k, ε) t-non-malleable extractor *if for every* (n,k) *source* X *and all functions* $f_1, \ldots, f_t\colon \{0,1\}^d \to \{0,1\}^d$ *with no fixed-points*[3] *it holds that:*

$$(\mathsf{nmExt}(X,Y), \mathsf{nmExt}(X, f_1(Y)), \ldots, \mathsf{nmExt}(X, f_t(Y), Y) \approx_\varepsilon (U_m, \mathsf{nmExt}(X, f_1(Y)), \ldots, \mathsf{nmExt}(X, f_t(Y), Y),$$

where Y *is the uniform distribution over* $\{0,1\}^d$*, independent of* X*.* ◊

[3] That is, for every i and every x, we have $f_i(x) \neq x$.

The initial motivation for non-malleable extractors was handling active adversaries in privacy amplification protocols (see, e.g., [DW09, DLWZ14]). Since then, non-malleable extractors proved to be extremely useful for constructing multiple source extractors [Li12b, Li13b, CZ16] and in fact, all constructions of two-source extractors for low min-entropy use non-malleable extractors as a main ingredient. We will talk about this intimate connection in the following chapters.

A t-non-malleable extractor can be used to generate a distribution which is almost t-wise independent in most of its coordinates. This observation, given in [CZ16], will be useful to us.

Lemma 2.4.2 ([CZ16])**.** *Let* $\mathsf{nmExt}\colon \{0,1\}^n \times [D] \to \{0,1\}$ *be a* (k,ε) *t-non-malleable extractor. Let* X *be any* (n,k) *source. Let* Bad *be the set defined by*

$$\mathsf{Bad} = \{r \in [D] \mid \exists \text{ distinct } r_1,\ldots,r_t \in [D], \forall i \in [t]\ r_i \neq r, |(\mathsf{nmExt}(X,r), \mathsf{nmExt}(X,r_1),\ldots,\mathsf{nmExt}(X,r_t)) - (U_1, \mathsf{nmExt}(X,r_1),\ldots,\mathsf{nmExt}(X,r_t))| > \sqrt{\varepsilon}\}. \quad (2.3)$$

Then, $\rho(\mathsf{Bad}) \leq \sqrt{\varepsilon}$*. In particular,* $R = [D] \setminus \mathsf{Bad}$ *is large,* $|R| \geq (1-\sqrt{\varepsilon})D$ *and for any* $r_1,\ldots,r_t \in R$*,*

$$(\mathsf{nmExt}(X,r_1),\ldots,\mathsf{nmExt}(X,r_t)) \approx_{t\sqrt{\varepsilon}} U_t. \quad (2.4)$$

We remark that the property in Equation (2.3) is stronger than the one in Equation (2.4). The second one says rows in R are almost t-wise independent. The first one says every row in R is independent from any other t rows – good or bad.

A successful line of work was devoted to constructing explicit non-malleable extractors [DLWZ14, CRS14, Li12a, Li12b, Li15a, CGL16, Coh16c, Coh16d, CL16, Coh16b, Coh17, Li17, Li18]. The current best explicit construction is given by Li [Li18].

Theorem 2.4.5 ([Li18, Coh16c])**.** *There exists a constant* $c_{\mathsf{nm}} \geq 2$ *such that for all integers* n,k,t *and every* $\varepsilon > 0$ *such that* $n \geq k \geq c_{\mathsf{nm}} t^2(\log n + \log\frac{1}{\varepsilon}\cdot\log\log\frac{1}{\varepsilon})$*, there exists an explicit* (k,ε) *t-non-malleable extractor*

$$\mathsf{nmExt}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

with $m = \frac{k}{3t}$ *output bits and seed-length* $d = c_{\mathsf{nm}} t^2(\log n + \log\frac{1}{\varepsilon}\cdot\log\log\frac{1}{\varepsilon})$*. The extractor is computable in time* $\mathsf{poly}(n,\log(1/\varepsilon))$*.*

2.5 Two-Source Extractors and Ramsey Graphs

In this section we complement the discussion about two-source extractors in the introduction with formal definitions and then discuss the weaker notion of Ramsey graphs.

2.5.1 Two-source extractors

Definition 2.5.1. *A function* $2\mathsf{Ext}\colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}$ *is an* $((n_1,k_1),(n_2,k_2),\varepsilon)$ two-source extractor *if for any two independent sources* X *and* Y*, where* X *is an* (n_1,k_1) *source and* Y *is an* (n_2,k_2) *source, the output distribution* $2\mathsf{Ext}(X,Y)$ *is* ε*-close to uniform. When* $k_1 = k_2 = k$ *we say the extractor is a two-source extractor for min-entropy* k*.* $\Diamond$

The existence of a two-source extractor for any min-entropy $k = \Omega(\log(n/\varepsilon))$ with $m = 2k - O(\log(1/\varepsilon))$ output bits was proved in [CG88]. In the same paper, an explicit construction of a two-source extractor for min-entropy $k > n/2$ was obtained. Remarkably, despite much attention [Bou05, Raz05, BSRZ15] and progress on relaxed settings [BKS+10, Rao09a, Li13b, Li13a, Li15b, Coh16a], the problem of constructing two-source extractors even for min-entropy as high as $k = 0.49n$ with $m = 1$ output bits remained open for 30 years until the celebrated result of Chattopadhyay and Zuckerman [CZ16] supporting $k = \mathsf{polylog}(n)$ (see also [Li16, Mek17]). In Chapter 3 we will overview the [CZ16] result and discuss our improved construction.

2.5.2 Ramsey graphs and two-source dispersers

Ramsey theory studies inevitable order that appears in large structures. It was initiated by Ramsey [Ram30], who showed that any graph over $N = 2^n$ vertices must contain a clique or an independent set of size $n/2$. A graph over N vertices is called *K-Ramsey* if it contains neither a clique nor an independent set of size K.

Inaugurating the probabilistic method, Erdös [Erd47] showed that there are $2n$-Ramsey graphs. He raised the challenge of giving an explicit description of such a graph and offered a bounty of \$100 for an explicit construction of an $O(n)$-Ramsey graphs. A related challenge is that of constructing a *K-Ramsey bipartite graph*, i.e., a bipartite graph with no bipartite clique or bipartite independent set of size K. Any explicit bipartite Ramsey graph can be translated into an explicit (non-bipartite) Ramsey graph with about the same parameters (see [Sha11]).

From a computer science point of view, bipartite Ramsey graphs are equivalent to *two-source dispersers* outputting one bit.

Definition 2.5.2. *A function* $\mathsf{2Disp}\colon [N] \times [N] \to \{0,1\}$ *is a (zero-error)* two-source K disperser *if for every two sets* $A, B \subseteq [N]$ *of cardinality at least* K, $\mathsf{Disp}(A,B) = \{0,1\}$. $\diamondsuit$

Such a disperser gives rise to a K-Ramsey bipartite graph with N vertices on each side. To appreciate the difficulty of constructing two-source extractors, note that:

Claim 2.5.3. *Every two-source extractor for min-entropy k with any non-trivial error readily implies a two-source K disperser, and thus also a bipartite K-Ramsey graph.*

Erdös's yet unmatched challenge initiated a line of beautiful constructions of Ramsey graphs [Abb72, Nag75, Fra77, Chu81, FW81, Alo98, Gro01]. The study of *pseudorandomness* gave a new perspective on the problem and led to exciting new constructions [Nao92, Bar06, BKS+10, BRSW12, Coh16e]. Cohen's construction [Coh16e] works for $K = 2^{\mathsf{poly}(\log\log N)}$, or alternatively $k = \mathsf{polylog}(n)$, matching the two-source extractor construction of [CZ16] that came shortly after. In fact, *all* Ramsey graphs following [Coh16e] construction are also two-source extractors.

Chapter 3

Constructing Two-Source Extractors – an Entropy-Efficient Reduction to Non-Malleable Extractors

The breakthrough result of Chattopadhyay and Zuckerman [CZ16] gives a *reduction* from the construction of explicit two-source extractors to the construction of explicit non-malleable extractors. However, even assuming the existence of optimal explicit non-malleable extractors only gives a two-source extractor (or a Ramsey graph) for $\mathsf{polylog}(n)$ entropy, rather than the optimal $O(\log n)$.

In this chapter, we first briefly give an overview of the [CZ16] reduction and then modify it to solve the above barrier. Using the currently best explicit non-malleable extractors [Li18] we get an explicit two-source extractor for n-bit sources having $k = (\log n)^{1+o(1)}$ min-entropy, which as we saw readily implies K-Ramsey bipartite graphs over N vertices on each side where $\log K = (\log \log N)^{1+o(1)}$. Any further improvement in the construction of non-malleable extractors would immediately yield a corresponding improved two-source extractor.

Intuitively, Chattopadhyay and Zuckerman use an extractor as a sampler, and we observe that one could use a weaker object – a *somewhere-random condenser* with a small entropy gap and a very short seed. We also show how to explicitly construct this weaker object using the error reduction technique of Raz, Reingold and Vadhan [RRV99], and the constant-degree dispersers of Zuckerman [Zuc07] that also work against extremely small tests.

3.1 Introduction

Recall that a graph G is a K-Ramsey bipartite graph if it contains neither bipartite cliques nor bipartite independent sets of size K. In Section 2.5 we also introduced the stronger notion of a two-source extractor and the holy-grail is constructing explicit two-source extractors for entropies $k = O(\log n)$, even with any non-trivial error guarantee. This would give K-Ramsey graphs for $K = \mathsf{poly}(n)$. The main result in this chapter is a construction that gets very close to optimal.

Theorem 3.1.1. *For every large enough n, there exists an explicit, constant-error, two-source extractor* $\mathsf{2Ext}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *supporting min-entropy* $k = (\log n)^{1+o(1)}$.

The [CZ16] two-source extractor construction is by a reduction. They show a reduction from the existence of explicit two-source extractors to the the construction of explicit non-malleable extractors. However, as noted by Cohen and Schulman [CS16] (and as we shall soon see), this reduction is not optimal in the sense that even if the explicit non-malleable extractor is optimal and has seed-length $O(\log \frac{n}{\varepsilon})$, the resulting two-source extractor is not optimal and requires both sources to have $\mathsf{polylog}(n)$ entropy rather than the optimal $O(\log n)$.

The reduction in this chapter solves this bottleneck. Specifically, we show that if one manages to construct *non-malleable extractors* with seed-length $f(n,\varepsilon)$ and entropy requirement $O(f(n,\varepsilon))$ then, essentially, this implies a constant-error *two-source extractor* with an entropy requirement of $O(f(n,\frac{1}{\mathsf{poly}(n)}))$. This, in particular, means that if one manages to construct optimal non-malleable extractors with $f(n,\varepsilon) = O(\log \frac{n}{\varepsilon})$ then this implies a two-source extractor with entropy requirement $O(\log n)$.[1]

Indeed, following our work, Cohen [Coh16b] and Li [Li17, Li18] constructed better explicit non-malleable extractors with the current best construction [Li18] having

$$f(n,\varepsilon) \;=\; O\left(\log n + \frac{\log\log(1/\varepsilon)}{\log\log\log(1/\varepsilon)} \log\frac{1}{\varepsilon}\right).$$

With these non-malleable extractors constructions, Cohen and Li invoked the reduction presented in this chapter and concluded a corresponding two-sources extractor. The two-source extractor one gets using Li's explicit non-malleable support

$$k = O\left(\log n \frac{\log\log n}{\log\log\log n}\right)$$

min-entropy. We stress, however, that the reduction presented here keeps working even if, or perhaps when, optimal non-malleable extractors would be obtained. Thus, the main contribution is the entropy-efficient reduction, translating any good non-malleable extractor to a corresponding two-source extractor without paying the $\mathsf{polylog}(n)$ penalty imposed by the [CZ16] construction.

3.1.1 The [CZ16] construction and its bottleneck

Let us briefly overview the [CZ16] construction. We are given $x_1, x_2 \in [N]$ sampled from two independent distributions X_1 and X_2, with min-entropies k_1 and k_2, respectively. We take a (k_1,ε_1) t-non-malleable extractor $\mathsf{nmExt}\colon [N]\times[D]\to\{0,1\}$ and write a $D\times 1$ table NM, where the rows are indexed by seeds $y\in[D]$, and in row y we write $\mathsf{NM}[y] = \mathsf{nmExt}(x_1,y)$. By the properties of non-malleable extractors there exists a large subset of the rows such that when we consider the distribution induced by these rows for a random $x_1 \sim X_1$, it is close to being t-wise independent (see Lemma 2.4.2). We deem every row in this subset "good" while the rest are "bad".

[1] We rely on a reduction of Cohen [Coh16c], allowing to transform a non-malleable extractor to a t-non-malleable extractor, and for a constant t we only pay in constant factors.

At this stage we would have liked to output $f(\mathsf{NM}[1],\ldots,\mathsf{NM}[D])$, for some *resilient function* f that is willing to accept a few bad players, and good players that are only close to being t-wise independent. Of course, since no one-source extractor exists – there is no such function f. Nevertheless, Chattopadhyay and Zuckerman explore why this approach fails. Since we are trying to do the impossible (or, rather, understand why the impossible is not possible), in our examination we shall assume the underlying extractor nmExt has optimal parameters.

Take a t-non-malleable extractor $\mathsf{nmExt}\colon [N]\times[D]\to\{0,1\}$ for $t=\mathsf{polylog}(n)$ and error ε_1. We get a table $\mathsf{NM}[y]=\mathsf{nmExt}(x_1,y)$ with D rows where any t good rows are $O(t\gamma)$-close to uniform for some $\gamma\geq\varepsilon_1$. Also, the number of bad rows is βD for some $\beta\geq\varepsilon_1$. We may choose any β and γ such that $\beta\gamma=\varepsilon_1$ and in particular we may take $\beta=\gamma=\sqrt{\varepsilon_1}$. If we take a non-malleable extractor with seed-length dependence $d=O(\log\frac{n}{\varepsilon_1})$ and $\varepsilon_1\leq\frac{1}{n}$, then $D=\mathsf{poly}(\frac{1}{\varepsilon_1})$ and $q=\beta D=\sqrt{\varepsilon_1}D\leq D^{1-\alpha}$ for some constant $\alpha>0$. To summarize, we get a table with D rows, at most $q=D^{1-\alpha}$ bad players for some $\alpha>0$, and every t good players are $t\sqrt{\varepsilon_1}$-close to uniform.

Recall that a function f is (q,t) resilient if it is resilient even when there are q bad players and the good players are only t-wise independent. Non-explicitly it is known that there are such functions for $q=D^{1-\alpha}$ bad players out of the D players and $t=\mathsf{polylog}(n)$. In fact, a large part of the [CZ16] paper is devoted to explicitly constructing such a function.

The two preceding paragraphs together imply that the table NM is close to a game with D players, where the good players are t-wise independent and the number of bad players q is at most $D^{1-\alpha}$, and f is a function resilient to such a situation. Hence, it seems, $f(\mathsf{NM}[1],\ldots,\mathsf{NM}[D])$ is close to uniform yielding an impossible one-source extractor.

Yet, there are no one-source extractors, and this is because there is a gap between what we proved about the table NM, and what we require from the (q,t) resilient function f. Specifically, we proved every t good rows are $t\sqrt{\varepsilon_1}$-close to uniform, but the function f assumes the good rows are *perfectly* t-wise independent. It *is* true that any distribution over D bits such that every t rows are ζ-close to uniform is $D^t\zeta$-close to a t-wise independent distribution ([AGM03], see Lemma 2.1.11) but in our case $D^t\zeta\gg 1$ because $D\geq\frac{1}{\varepsilon_1^2}\geq\frac{1}{\zeta^2}$. Thus, the impossible does not happen and the one-source construction fails.

Chattopadhyay and Zuckerman use the other source to bypass the above problem. They use X_2 to sample rows from the table NM, i.e., they take a (k_2,ε_2) extractor $\mathsf{Ext}\colon [N]\times[R]\to[D]$ and output

$$\begin{aligned}\mathsf{2Ext}(x_1,x_2) &= f(\mathsf{NM}[\mathsf{Ext}(x_2,1)],\ldots,\mathsf{NM}[\mathsf{Ext}(x_2,R)])\\ &= f(\mathsf{nmExt}(x_1,\mathsf{Ext}(x_2,1)),\ldots,\mathsf{nmExt}(x_1,\mathsf{Ext}(x_2,R))).\end{aligned}$$

In other words, x_2 samples R rows from the table NM, and these samples are fed into the resilient function. We will soon see how the sample helps solving the problem we had before.

Since extractors are good samplers, if k_1 is large enough, almost all x_2-s sample well. Namely, the fraction of the bad players in the R sampled rows is about $\sqrt{\varepsilon_1}+\varepsilon_2$ and each t good players are $t\sqrt{\varepsilon_1}$-close to uniform. We take $\varepsilon_1\ll\varepsilon_2$, so we can just think of $\sqrt{\varepsilon_1}+\varepsilon_2$ as ε_2 fraction of bad rows. If we take a small enough ε_2 and an extractor with seed-length $O(\log\frac{n}{\varepsilon_2})$ we again get that, with high probability, the sample contains at most $R^{1-\alpha}$ bad players out of the R players. Also, as before, every t good players are $t\sqrt{\varepsilon_1}$-close to uniform.

Therefore, again, we may conclude that the good rows in R are $(R^t \cdot t\sqrt{\varepsilon_1})$-close to being truly t-wise independent. Now, however, we may choose ε_1 *smaller* than R^t so then $R^t \cdot t\sqrt{\varepsilon_1} < 1$ and the argument goes through.

In a nutshell, with one source D is a function of ε_1 and $D^t\varepsilon_1$ is necessarily larger than 1; with two sources ε_1 may be chosen way smaller than R^t and the argument works!

As beautiful as it is, the argument has its own limitations. We first argue that the number of bad rows in the sampled table is at least $\sqrt{R}$ (out of the R rows in the table). To see this recall that samplers are (almost) equivalent to extractors (see Section 2.2.1), and an extractor with error ε_2 has seed length $d_2 \geq 2\log(\frac{1}{\varepsilon_2})$. Thus, the number of rows R is at least $\frac{1}{\varepsilon_2^2}$ and the number of bad rows is at least $\varepsilon_2 R \geq \sqrt{R}$. All the currently known (q,t) resilient functions that handle $q \geq \sqrt{R}$ bad players require t which is poly-logarithmic in R. The required entropy from X_1 is at least the entropy required by the t-non-malleable extractor nmExt to output one bit, which is clearly at least t. Altogether, this implies that the [CZ16] construction requires $k_1 = k_2 = \mathsf{polylog}(n)$, and this is true even if the non-malleable extractor has optimal seed-length $O(\log \frac{n}{\varepsilon_1})$.

We note that ideas of converting a weak source into a table (or, a somewhere-random source) were already applied in Rao's work [Rao09a] and techniques for obtaining non-oblivious bit-fixing sources (i.e., t-wise independence in a large fraction of the rows) were developed, using alternating extraction protocols, in Li's works [Li13a, Li15b].

3.1.2 An overview of our construction

Cohen and Schulman [CS16] note that all previous explicit multi-source extractors (or dispersers) work with entropy at least $\log^2 n$. They were able to construct a multi-source extractor requiring only $(\log n)^{1+o(1)}$ entropy using a new primitive called independence-preserving merger. In a subsequent paper, Cohen [Coh16b] constructs a five-source extractor for entropy $(\log n)^{1+o(1)}$.[2] The new ingredient that allows this lower entropy requirement is that the resilient function that is used is the Majority function, and Viola [Vio14] showed the majority function is $(q = D^{\frac{1}{2}-\alpha}, t = O(1))$-resilient, i.e., it suffices that the good players are t-wise independent for some constant t! However, to be able to use the Majority function the number q of bad players has to be below square root the total number of players, and, informally speaking, Cohen uses the other four sources to guarantee that the number of bad rows is at most $R^{0.4}$.

The starting point of the current work is the observation that condensers with a small entropy gap (that we soon define) are good samplers and their dependence on ε can get as small as $1 \cdot \log(\frac{1}{\varepsilon})$. Let us recall the definition of a condenser and define its entropy gap.

Definition 3.1.2. *A function* $\mathsf{Cond}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* $(k \to k', \varepsilon)$ *condenser, if for every* (n,k) *source* X, $\mathsf{Cond}(X, U_d)$ *is* ε*-close to an* (m,k') *source. The* entropy loss *of the condenser is* $k + d - k'$ *and the* entropy gap *is* $m - k'$. ◊

Dodis et al. [DPW14] observe that even for entropy gap as small as 1, non-explicit constructions achieve entropy loss which is only $\log\log(\frac{1}{\varepsilon}) + O(1)$ compared to an entropy

[2] In fact, four sources suffice for the construction, because the fourth source is redundant as the advice correlation breaker works also with a weak dense seed.

loss of $2\log(\frac{1}{\varepsilon})$ when there is no entropy gap. Furthermore, [DPW14] show that condensers with small entropy gap are still good samplers when the test set is small (see Lemma 3.3.4). Dodis et al. use this property for key derivation without entropy waste. They also show that in non-explicit constructions the seed-length dependence on the error is $1 \cdot \log(\frac{1}{\varepsilon})$ rather than the $2 \cdot \log(\frac{1}{\varepsilon})$ in extractors.

As condensers with a small entropy gap are good samplers against small tests, it is not difficult to see that one can replace the extractor in [CZ16] with such a condenser, and everything stays (almost) the same. Now assume there is an *explicit* construction of a small entropy gap condenser that has optimal dependence of the seed-length d on the error ε. Then, we may take $d < 2\log(\frac{1}{\varepsilon})$ and use the Majority function, which implies we can work with a constant t rather than a poly-logarithmic t. Assuming we also have an explicit t-non-malleable extractor with optimal seed-length $O(\log \frac{n}{\varepsilon})$ we get a two-source extractor requiring entropy $O(\log n)$, as desired.

We see this observation as the main *conceptual* idea of the chapter, namely, that one can replace the sampler in the [CZ16] construction with a sampler against small tests, and by doing so, at least theoretically, one may reduce the independence requirement to a constant t, and the entropy requirement to the optimal $O(\log n)$. Incidentally, this framework also *simplifies* the proof, as the bulk of the work in [CZ16] is devoted to explicitly constructing an explicit (q, t) resilient function, which can now be replaced by the Majority function.

Unfortunately, we are not aware of an *explicit* construction of condensers achieving a small entropy gap and seed-length less than $2\log(\frac{1}{\varepsilon})$. We remark that while the seed-length of the condenser of [GUV09] approaches $\log(\frac{1}{\varepsilon})$, its entropy gap is big.

The main *technical* primitive of the chapter is an explicit construction of a *somewhere-random* condenser with a short seed and a small entropy gap, and showing such an object also suffices for the reduction. Together with the pretty good explicit non-malleable extractor constructions we currently have, this gives an explicit two-source extractor with nearly logarithmic entropy requirement.

Recall that a somewhere-random condenser is a weaker object than a condenser. The output of a somewhere random condenser is divided into *blocks*, and, roughly speaking, the guarantee is that one of these output blocks is close to having high min-entropy. A condenser is a somewhere-random condenser with just one block (for the exact definitions, refer to Section 2.3.3).

We construct a somewhere-random condenser with a constant number of blocks, very small entropy gap, and seed-length $(1 + \alpha)\log(\frac{1}{\varepsilon})$, for any constant $\alpha < 1$. The idea is to start with an extractor that has the wrong dependence on the error, and decrease its error to the desired value ε in an efficient way that gives dependence smaller than $2\log(\frac{1}{\varepsilon})$. We use the error reduction scheme suggested by Raz et al. [RRV99] that works by sampling (with a sampler) a constant number of seeds and outputting all the corresponding outputs of the initial extractor. They show the obtained output is a somewhere-random source with a small entropy gap and a very low error.

The sampler used in [RRV99] is obtained by taking a random walk (of constant length) over an expander. The analysis shows that such a reduction has dependence at least $2\log(\frac{1}{\varepsilon})$ on the error. Instead, we observe that what is needed in the reduction is a disperser against very small tests (i.e., the image of any large enough set is not contained in any small set). The fact that we only need to handle small tests is again crucial, as we saw Zuckerman [Zuc07]

already constructed such dispersers having a constant degree! Usually, the degree has to be logarithmic, but for the special parameters that we need that reflect the fact we only need to handle very small tests, the degree may be constant. Using these dispersers in the error reduction scheme, we get the desired somewhere random condensers. As a by-product, we obtain a slight simplification and a generalization of the [RRV99] error reduction scheme, which we point out in Theorem 3.2.5.

Armed with that we go back to the [CZ16] construction and replace the extractor Ext with a somewhere-random condenser. As we use a somewhere-random condenser rather than a condenser, when x_2 samples the rows of NM, we get an $R \times A$ table (rather than the $R \times 1$ table we had previously) where A is the constant number of blocks of our somewhere-random condenser. Let us say a row is good if one of its A blocks is good. The property of the table is that the number of bad rows is at most $R^{0.4}$, and the good rows are, informally speaking, t-wise somewhere random. Here we apply another trick from [Coh16b]: We take the parity of each row and apply the resilient function on it. The result is an almost balanced bit and we are done.

3.2 Low-Error S.R. Condensers with a Short Seed and a Small Entropy Gap

In this section we construct s.r. condensers with ε error that have seed-length $(1+\alpha)\cdot\log(\frac{1}{\varepsilon})$ and entropy gap $O(\log\frac{1}{\varepsilon})$.

Theorem 3.2.1. *For every constant $0 < \alpha < 1$ there exists a constant A such that for every integer n, $0 < \varepsilon \leq (\frac{1}{n})^{\frac{4c_{\mathsf{GUV}}}{\alpha}}$ and every integer $m \leq \frac{n}{2} - \log(\frac{1}{\varepsilon})$ there exists an explicit function*

$$\mathsf{SRC}\colon [N] \times [R'] \times [A] \to [M]$$

that is a $(k = 2m + \log(\frac{1}{\varepsilon}) \to m - 2\log(\frac{1}{\varepsilon}) - O(a), \varepsilon)$ s.r. condenser with $r' = (1+\alpha)\log(\frac{1}{\varepsilon})$.

Notice that the s.r. condenser achieves the small error ε using only a constant number of blocks, a small entropy gap (i.e., the min-entropy in the s.r. source is close to the block length) and seed-length close to $\log(\frac{1}{\varepsilon})$.

Proof. Fix $\alpha > 0$.

Construction: The first ingredient is an extractor $\mathsf{Ext}\colon [N] \times [R] \to [M]$ that has error ε_0 that is too high for us, $\varepsilon_0 = \varepsilon^{1/c}$, but seed-length that is still within our budget, say, $\frac{\alpha}{2}\log(\frac{1}{\varepsilon})$. Our goal is to reduce the error to $\varepsilon = \varepsilon_0^c$. Specifically, set $\varepsilon_0 = \varepsilon^{\frac{\alpha}{4c_{\mathsf{GUV}}}}$ and notice that $\varepsilon_0 \leq \frac{1}{n}$. By Theorem 2.2.3, there exists

$$\mathsf{Ext}\colon [N] \times [R] \to [M]$$

that is an explicit strong $(2m, \varepsilon_0)$ seeded extractor with

$$r = c_{\mathsf{GUV}} \log\frac{n}{\varepsilon_0} \leq 2c_{\mathsf{GUV}} \log\frac{1}{\varepsilon_0} = \frac{\alpha}{2}\log\frac{1}{\varepsilon}.$$

The second ingredient is a disperser, that we use as a sampler to sample many (dependent) seeds of Ext, one of which is good. Specifically, set $r' = (1+\alpha)\log(\frac{1}{\varepsilon})$ and take

$$\Gamma\colon [R'] \times [A] \to [R]$$

to be the $\left(K = (R')^{\frac{\alpha}{1+\alpha}}, K' = 2\varepsilon_0 R\right)$ disperser guaranteed by Theorem 2.3.4, when plugging-in $a = \frac{\alpha}{1+\alpha}$ and $b = \frac{1}{2}$. Notice that $K^{\frac{1}{2}} = (R')^{\frac{1}{2}\frac{\alpha}{1+\alpha}} = (\frac{1}{\varepsilon})^{(1+\alpha)\frac{1}{2}\frac{\alpha}{1+\alpha}} = (\frac{1}{\varepsilon})^{\frac{\alpha}{2}} \geq R$. The degree is then

$$A = O\left(\frac{r'}{\log\frac{1}{2\varepsilon_0}}\right) = O\left(\frac{(1+\alpha)\log\frac{1}{\varepsilon}}{\frac{\alpha}{4c_{\mathsf{GUV}}}\log\frac{1}{\varepsilon} - 1}\right) = O(1).$$

We define $\mathsf{SRC}\colon [N] \times [R'] \times [A] \to [M]$ by:

$$\mathsf{SRC}(x, y', z) \;=\; \mathsf{Ext}(x, \Gamma(y', z)).$$

Correctness: Let X be a k-source for $k = 2m + \log(\frac{1}{\varepsilon})$. Without loss of generality, X is flat (because otherwise it is a convex combination of such sources). Our goal is to prove that $\mathsf{SRC}(X, U_{r'})$ is a s.r. source with $k' = m - 2\log(\frac{1}{\varepsilon}) - O(a)$ min-entropy.

Since Ext is an extractor, the output distribution $\mathsf{Ext}(X, U_r)$ is ε_0-close to uniform. Define the set of Δ-*heavy elements* in $[M]$ by:

$$\mathbf{H} = \left\{w \in [M] : \Pr[\mathsf{Ext}(X, U_r) = w] \geq \frac{\Delta}{M}\right\}.$$

We claim:

Claim 3.2.2. *For $\Delta > 2$, $|\mathbf{H}| < \frac{2\varepsilon_0}{\Delta} M$.*

Proof. Notice that $|\mathbf{H}| \cdot \frac{\Delta}{M} \leq \Pr[\mathsf{Ext}(X, U_r) \in \mathbf{H}] \leq \frac{|\mathbf{H}|}{M} + \varepsilon_0$, where the upper bound follows because $\mathsf{Ext}(X, U_r)$ is ε_0-close to uniform. Thus $\frac{|\mathbf{H}|}{M}(\Delta - 1) \leq \varepsilon_0$ and $|\mathbf{H}| \leq \frac{\varepsilon_0}{\Delta - 1} M < \frac{2\varepsilon_0}{\Delta} M$. □

We will work with

$$\Delta = \frac{4A}{\varepsilon} \geq 4.$$

There are two possible ways $\mathbf{H}$ may get its weight in Ext, and they react differently to amplification. First, there are "typical" elements for which the fraction of seeds falling into $\mathbf{H}$ is about right (the density of $\mathbf{H}$ plus Ext's error which is ε_0). In this case the amplification with the disperser Γ guarantees that we miss $\mathbf{H}$ with one of our samples with good probability.

Moreover, there are "bad" elements $x \in X$ for which many seeds y fall into $\mathbf{H}$. Bad elements do not react well to amplification (e.g., amplification has no effect when all seeds fall into $\mathbf{H}$) but there are very few of them. Formally, we define the set of bad inputs $x \in \mathrm{Supp}(X)$ by

$$\mathsf{Bad}_X = \left\{x \in [N] : \Pr_{y \sim U_r}[\mathsf{Ext}(x, y) \in \mathbf{H}] \geq \left(1 + \frac{2}{\Delta}\right)\varepsilon_0\right\}.$$

Claim 3.2.3. $|\mathsf{Bad}_X| < 2^{2m}$.

Proof. Suppose $|\mathsf{Bad}_X| \geq 2^{2m}$. Let B be uniformly distributed over Bad_X. Then, $\mathsf{Ext}(B, U_r)$ is ε_0-close to uniform, and therefore

$$\left(1 + \frac{2}{\Delta}\right)\varepsilon_0 \leq \Pr_{x\sim B, y\sim U_r}[\mathsf{Ext}(x,y) \in \mathbf{H}] \leq \frac{|\mathbf{H}|}{M} + \varepsilon_0.$$

But then $\frac{2\varepsilon_0}{\Delta} \leq \frac{|\mathbf{H}|}{M}$, which contradicts to the previous claim. □

Now consider a "typical" element, i.e. $x \in \mathrm{Supp}(X) \setminus \mathsf{Bad}_X$. As $x \notin \mathsf{Bad}_X$ there are only a few seeds y (about ε_0) such that $\mathsf{Ext}(x,y)$ falls into $\mathbf{H}$. If we sample a constant number of independent seeds, then except for probability $\varepsilon = \varepsilon_0^{O(1)}$, one of them will *not* fall into $\mathbf{H}$, and we get a s.r. source (but with a long random seed). Raz et al. [RRV99] replace the independent samples with a good sampler (a random walk on expanders). We use a different sampler – Zuckerman's sampler from Theorem 2.3.4, because we are in the small-test regime in which Zuckerman's sampler has a constant degree.

Formally, let I be a random variable defined as follows. For $x \in [N]$ and $y' \in [R']$, $I(x,y')$ is an arbitrary $z \in [A]$ such that $\mathsf{Ext}(x, \Gamma(y',z)) \notin \mathbf{H}$ if such a z exists, and 0 otherwise. Let I' be the same as I except that for all z with $\Pr[I = z] \leq \frac{4}{\Delta}$, if $I(x,y') = z$ we let $I'(x,y') = 0$.

Claim 3.2.4.

- $\Pr[I = 0] \leq 2\varepsilon$.
- $\Pr[I' = 0] \leq 3\varepsilon$.
- *For every* $z \in [A]$, $H_\infty(\mathsf{Ext}(X, \Gamma(U_{r'}, I'))|I' = z) \geq m - 2\log\Delta + 2$.

Proof. For the first item, $\Pr[I = 0] \leq \Pr[X \in \mathsf{Bad}_X] + \Pr[I = 0 | X \notin \mathsf{Bad}_X]$.

- Clearly, $\Pr[X \in \mathsf{Bad}_X] \leq |\mathsf{Bad}_X| \cdot 2^{-k} = |\mathsf{Bad}_X| \cdot 2^{-(2m + \log(\frac{1}{\varepsilon}))} \leq \varepsilon$. Intuitively, we "drown" the bad elements among all the elements in X. Said differently, we increase the entropy requirement to reduce the error.
- Fix an element $x \notin \mathsf{Bad}_X$ and call $y \in [R]$ *bad for* x if $\mathsf{Ext}(x,y) \in \mathbf{H}$. As $x \notin \mathsf{Bad}_X$, the number of seeds y that are bad for x is at most $(1 + \frac{2}{\Delta})\varepsilon_0 R \leq 2\varepsilon_0 R$. Notice that $(I = 0 | X = x)$ if $y' \in [R']$ is such that $\Gamma(y',1), \ldots, \Gamma(y',A)$ are all bad for x. Since Γ is a
$$\left(K = (R')^{\frac{\alpha}{1+\alpha}}, K' = 2\varepsilon_0 R\right)$$
disperser, the number of such y'-s is at most $K = (R')^{\frac{\alpha}{1+\alpha}}$. Hence,
$$\begin{aligned}\Pr[I = 0 | X \notin \mathsf{Bad}_X] &\leq \frac{(R')^{\frac{\alpha}{1+\alpha}}}{R'} = (R')^{-(1-\frac{\alpha}{1+\alpha})} \\ &= (R')^{-(\frac{1}{1+\alpha})} = (\varepsilon^{1+\alpha})^{\frac{1}{1+\alpha}} = \varepsilon,\end{aligned}$$
as desired.

For the second item, $\Pr[I' = 0] \leq \Pr[I = 0] + \frac{4A}{\Delta} = 3\varepsilon$.

For the third item, let w be in the support of $(\mathsf{Ext}(X, \Gamma(U_{r'}, z)) \mid I' = z)$. Hence $w \notin \mathbf{H}$. It follows that

$$\begin{aligned}\Pr[\mathsf{Ext}(X, \Gamma(U_{r'}, z)) = w | I' = z] &\leq \frac{\Pr[\mathsf{Ext}(X, \Gamma(U_{r'}, z)) = w]}{\Pr[I' = z]} \\ &= \frac{\Pr[\mathsf{Ext}(X, U_r) = w]}{\Pr[I' = z]} \\ &\leq \frac{\Delta}{M} \cdot \frac{1}{\Pr[I' = z]} \leq \frac{\Delta^2}{4M},\end{aligned}$$

where the equality on the second line follows from Claim 2.3.7. Thus, for every $z \in [A]$ in the support of I',

$$H_\infty(\mathsf{Ext}(X, \Gamma(U_{r'}, I')) \mid I' = z) \geq m - 2\log\Delta + 2,$$

concluding our proof. □

□

If one does not care about keeping the seed-length of the construction close to $1 \cdot \log(\frac{1}{\varepsilon})$, a merging step can be performed, leading to a generic error reduction scheme for condensers. Also, the case of reducing the error from ε_0 to a constant power of ε_0 is not the standard one, and we are usually interested, say, in obtaining a polynomially-small error condenser from a constant error one. Using the same techniques and the curve merger from Theorem 2.3.15, one obtains the following version of the [RRV99] result:

Theorem 3.2.5. *Suppose* $\mathsf{Cond} \colon [N] \times [D] \to [M]$ *is an explicit* $(k_{\text{in}} \to k_{\text{out}}, \varepsilon_0)$ *condenser,* $0 < \alpha < 1$ *and* $0 < \varepsilon < \varepsilon_0$*. Then, there exists an explicit* $\mathsf{Cond}' \colon [N] \times [D'] \to [M]$ *that is a* $(k'_{\text{in}} \to k'_{\text{out}}, \varepsilon)$ *condenser with*

- $k'_{\text{in}} = k_{\text{in}} + \log\frac{1}{\varepsilon}$,
- $k'_{\text{out}} = (1-\alpha)k_{\text{out}} - O\left(\log\frac{d}{\varepsilon(1-\varepsilon_0)}\right)$, *and*
- $d' = d + O(\frac{1}{\alpha}\log\frac{d}{\varepsilon})$.

The proof goes along the same lines as those of Theorem 3.2.1 and we omit it (and we also do not need Theorem 3.2.5 for our construction).

3.3 From S.R. Condensers to S.R. Samplers

As we discussed in Section 2.2.1, extractors are good samplers in the sense that if Ext is a (k, ε) seeded extractor then for every test S we have that $\Pr[\mathsf{Ext}(X, U) \in S]$ deviates from the density of S by at most ε. However, extractors are quite limited in the parameters they can achieve and in particular require seed-length that is at least $2\log(\frac{1}{\varepsilon})$ [RTS00]. Dodis,

Pietrzak and Wichs [DPW14] observed that if we are only interested in fooling sparse tests (or equivalently, if we also allow multiplicative error), it suffices to use condensers with a small entropy gap. Moreover, Dodis et al. note that such condensers can bypass the severe limitations that confine extractors. In this section we show that the Dodis et al. result carries over to s.r. condensers. We obtain explicit s.r. samplers with the parameters we need, and in particular seed-length that is smaller than $2\log(\frac{1}{\varepsilon})$.

We first define samplers with both multiplicative and additive error.

Definition 3.3.1. *Let* $S\colon [N] \times [R'] \to [D]$.

- *We say* $x \in [N]$ *is* (c,ε) *bad for* $B \subseteq [D]$ *if* $\Pr_{y' \in [R']}[S(x,y') \in B] > c\rho(B) + \varepsilon$.
- *We say* S *is a* $(K; c, \varepsilon)$ sampler *if for every* $B \subseteq [D]$,
$$|\{x \in [N] \mid x \text{ is } (c,\varepsilon) \text{ bad for } B\}| < K.$$

$\Diamond$

Definition 3.3.2. *Let* $S\colon [N] \times [R'] \times [A] \to [D]$.

- *We say* $x \in [N]$ *is* (c,ε) *bad for* $B \subseteq [D]$ *if* $\Pr_{y' \in [R']}[\forall_{z \in [A]}\ S(x,y',z) \in B] > c\rho(B) + \varepsilon$.
- *We say* S *is a* $(K; c, \varepsilon)$ s.r. sampler *if for every* $B \subseteq [D]$,
$$|\{x \in [N] \mid x \text{ is } (c,\varepsilon) \text{ bad for } B\}| < K.$$

$\Diamond$

Definition 3.3.3. *We say* $S\colon [N] \times [R'] \times [A] \to [D]$ *is* simple *if for every* $x \in [N]$, *and every* $y'_1, y'_2 \in [R']$, $z_1, z_2 \in [A]$, *if* $(y'_1, z_1) \neq (y'_2, z_2)$ *then* $S(x,y'_1,z_1) \neq S(x,y'_2,z_2)$. $\Diamond$

The following lemma is based on [DPW14].

Lemma 3.3.4. *If* X *is a* $(d, d-g)$*-source then for every set* $B \subseteq [D]$,
$$\Pr[X \in B] \ \leq\ 2^g \cdot \rho(B).$$

Proof. If X is flat, then the probability that X is in B is bounded by the density of B inside the support of X, i.e., it is at most $\frac{|B|}{\mathrm{Supp}(X)|} \leq \frac{|B|}{2^{d-g}} = 2^g \cdot \rho(B)$. Since every $(d, d-g)$ source is a convex combination of such flat sources, the lemma follows. □

Lemma 3.3.5. *If* $X = X_1 \circ \ldots \circ X_A$ *is a* $(d, d-g, \zeta)$ *s.r. source then for every set* $B \subseteq [D]$,
$$\Pr_{x \sim X}\ [\forall_{z \in [A]}\ X_z \in B] \ \leq\ 2^g \cdot \rho(B) + \zeta.$$

Proof. X is ζ-close to a $(d, d-g)$ s.r. source X'. Let I be an indicator of X'.

$$\begin{aligned}
\Pr[\forall_{z \in [A]}\ X'_z \in B] \ &\leq\ \sum_{i=1}^{A} \Pr[I=i] \cdot \Pr_{x \sim X'}\ [\forall_{z \in [A]}\ X'_z \in B \mid I = i] \\
&\leq\ \sum_{i=1}^{A} \Pr[I=i] \cdot \Pr\ [\ X'_i \in B \mid I = i] \\
&\leq\ \sum_{i=1}^{A} \Pr[I=i] \cdot 2^g \cdot \rho(B) \ \leq\ 2^g \cdot \rho(B),
\end{aligned}$$

where the third inequality follows from Lemma 3.3.4. □

Theorem 3.3.6. *If* $\mathsf{Cond}\colon [N] \times [R'] \times [A] \to [D]$ *is a* $(k \to d-g, \varepsilon)$ *s.r. condenser then* Cond *is a* $(2^k; 2^g, \varepsilon)$ *s.r. sampler.*

Proof. Let $B \subseteq [D]$, and let Bad denote the set of elements in $[N]$ that are $(2^g, \varepsilon)$ bad for B. If $|\mathsf{Bad}| \geq K$, then $\mathsf{Cond}(\mathsf{Bad}, U_{r'})$ is a $(d, d-g, \varepsilon)$ s.r. source. By Lemma 3.3.5,

$$\Pr_{x \in \mathsf{Bad},\ y' \in [R']} [\forall_{z \in [A]}\ \mathsf{Cond}(x, y')_z \in B] \ \leq\ 2^g \cdot \rho(B) + \varepsilon.$$

Therefore, there must exist at least one $x \in \mathsf{Bad}$ such that $\Pr_{y' \in [R']}[\forall_{z \in [A]}\ \mathsf{Cond}(x, y')_z \in B] \leq 2^g \cdot \rho(B) + \varepsilon$, in contradiction to the definition of Bad. Thus, $|\mathsf{Bad}| < K$, as required. □

We now instantiate Theorem 3.3.6 with the s.r. condenser from Theorem 3.2.1 to obtain:

Theorem 3.3.7. *For every constant* $0 < \alpha < 1$ *there exist constants* $A = A(\alpha), b = b(\alpha)$ *such that for every integer* n *and* $d \leq n/2$ *there exists an explicit*

$$S\colon [N] \times [R'] \times [A] \to [D]$$

that is a $(K = D^2; c = (\frac{1}{\varepsilon})^4, \varepsilon = (\frac{1}{n})^b)$ *simple s.r. sampler with* $r' = (1+\alpha)\log(\frac{1}{\varepsilon})$.

Proof. We are given α. Set A to be the constant $A = A(\alpha)$ given in Theorem 3.2.1. Set $b = b(\alpha) = \frac{4c_{\mathsf{GUV}}}{\alpha}$, $\varepsilon = (\frac{1}{n})^b$ and let $r' = (1+\alpha)\log(\frac{1}{\varepsilon})$. Given d, set $m = d - a - r'$. Let

$$\mathsf{SRC}\colon [N] \times [R'] \times [A] \to [M]$$

denote the $(2m + \log(\frac{1}{\varepsilon}) \to m - 2\log(\frac{1}{\varepsilon}) - O(a), \varepsilon)$ s.r. condenser from Theorem 3.2.1. Define a new condenser

$$S\colon [N] \times [R'] \times [A] \to [R'] \times [A] \times [M]$$

by

$$S(x, y', z) = (y', z, \mathsf{SRC}(x, y', z)).$$

It is immediate that S is simple (because the seed is part of the output). Notice that $M \cdot R' \cdot A = D$. By Theorem 3.3.6, S is a $(2^{2m+\log(\frac{1}{\varepsilon})}; c = 2^{m+r'+a-(m-2\log(\frac{1}{\varepsilon})-O(a))}, \varepsilon)$ s.r sampler, and:

- $2^{2m+\log(\frac{1}{\varepsilon})} = \frac{M^2}{\varepsilon} \leq (MR')^2 \leq D^2 = K$.
- $c = 2^{m+r'+a-(m-2\log(\frac{1}{\varepsilon})O(a))} \leq 2^{r'+2\log(\frac{1}{\varepsilon})+O(a)} = 2^{(3+\alpha)\log(\frac{1}{\varepsilon})+O(a)} \leq \varepsilon^{-4}$.

□

3.4 From S.R. Samplers to Two-Source Extractors

In this section we prove our main theorem for this chapter.

Theorem 3.4.1 (Theorem 3.1.1 restated)**.** *For every constant* $\varepsilon > 0$ *there exists a constant* c *such that for every large enough integer* n*, there exists an explicit* $((n, k_1), (n, k_2), \varepsilon)$ *two-source extractor* $2\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *for any* $k_1, k_2 \geq c \cdot \log n \cdot \frac{\log\log n}{\log\log\log n}$.

The proof closely follows the intuition given in the introduction, but makes it rigorous and as a result many parameters enter the discussion.

Proof.

The construction: We are given n and a constant ε.

- Set t to be large enough so that $c_{\mathsf{maj}}\frac{\log t}{t} \leq \frac{\varepsilon}{6}$, where c_{maj} is the constant from Lemma 2.1.14. A calculation shows that
$$t = \left\lceil \frac{36c_{\mathsf{maj}}^2}{\varepsilon^2} \right\rceil$$
suffices, so $t = O(1)$.
- We now fix parameters for the s.r. sampler. We set $\alpha = \frac{1}{2}$. Let $A = A(\alpha)$, $b = b(\alpha)$ be the constants determined by α in Theorem 3.3.7. Fix $\varepsilon_2 = (\frac{1}{n})^b$ and $c = (\frac{1}{\varepsilon_2})^4$. Set $R' = (\frac{1}{\varepsilon_2})^{1+\alpha}$.
- Set ε_1 small enough so that $c_{\mathsf{maj}}t\sqrt{\varepsilon_1}(R')^t \leq \frac{\varepsilon}{6}$ and $c\sqrt{\varepsilon_1} \leq \frac{1}{2}(R')^{-0.6}$. We may take $\varepsilon_1 = \frac{\varepsilon^2}{6^2c_{\mathsf{maj}}^2t^2c^2(R')^{2t}}$ and so $\varepsilon_1^{-1} = (cR')^{O(1)} = n^{O(1)}$.
- Set $d = c_{\mathsf{nm}}(tA)^2 f(n, \varepsilon_1)$ where
$$f(n, \varepsilon_1) = \log n + \frac{\log\log(1/\varepsilon_1)}{\log\log\log(1/\varepsilon_1)} \log\left(\frac{1}{\varepsilon_1}\right)$$
and c_{nm} is the constant from Theorem 2.4.5 (we note that everything keeps working if someone gives an improved construction with, say, $f(n, \varepsilon) = O(\log \frac{n}{\varepsilon})$).
- Let
$$S\colon [N] \times [R'] \times [A] \to [D]$$
be the $(K = D^2; c, \varepsilon_2)$ s.r. simple sampler guaranteed by Theorem 3.3.7 where N, R', D, A, c and ε_2 were defined before.
- Let
$$\mathsf{nmExt}\colon [N] \times [D] \to \{0,1\}$$
be the $(k_1 = d, \varepsilon_1)$ t'-non-malleable extractor guaranteed by Theorem 2.4.5 set to output one bit, for $t' = tA$. Notice that d was chosen to be sufficiently large for nmExt.

After fixing the above, given $x_1, x_2 \in [N]$, the construction is as follows:

1. For every $y' \in [R']$ and $z \in [A]$, compute
$$\mathsf{NM}(x_1, x_2; y', z) = \mathsf{nmExt}(x_1, S(x_2, y', z)).$$
2. For every $y' \in [R']$, compute
$$\oplus\mathsf{NM}(x_1, x_2; y') \;=\; \bigoplus_{z=1}^{A} \mathsf{NM}(x_1, x_2; y', z).$$

3. Output

$$\mathsf{2Ext}(x_1, x_2) \;=\; \mathrm{Maj}(\oplus\mathsf{NM}(x_1, x_2; 1), \ldots, \oplus\mathsf{NM}(x_1, x_2; R')).$$

Correctness: We now prove correctness. Let X_1 be an (n, k_1) source and X_2 be an (n, k_2) source for $k_2 = k + \log\frac{2}{\varepsilon}$ independent from X_1.

Let $\mathsf{Bad} \subseteq [D]$ be the set of bad rows of nmExt of density at most $\sqrt{\varepsilon_1}$ guaranteed to us by Lemma 2.4.2. Note that Bad depends only on X_1. We say $x_2 \in [N]$ is *bad* if x_2 is (c, ε_2) bad for Bad (see Definition 3.3.2) and good otherwise.

Claim 3.4.2. $\Pr_{x_2 \sim X_2}[x_2 \text{ is bad}] \leq \frac{\varepsilon}{2}$.

Proof. The number of bad elements is at most K, and $H_\infty(X_2) \geq k + \log\frac{2}{\varepsilon}$ so we can conclude that $\Pr_{x_2 \sim X_2}[x_2 \text{ is bad}] \leq \frac{K}{2^{k_2}} = \frac{\varepsilon}{2}$. □

Define

$$\oplus R(x_1, x_2) = (\oplus\mathsf{NM}(x_1, x_2, 1), \ldots, \oplus\mathsf{NM}(x_1, x_2, R')).$$

Lemma 3.4.3. *For every good $x_2 \in \mathrm{Supp}(X_2)$, $\oplus R(X_1, x_2)$ is a (q, t, γ) non-oblivious bit-fixing source for $q = (R')^{0.4}$ and $\gamma = t\sqrt{\varepsilon}_2$.*

Proof. Fix any good $x_2 \in \mathrm{Supp}(X_2)$. Call $y' \in [R']$ *a bad row* if for every $z \in [A]$, $S(x, y', z) \in \mathsf{Bad}$, and good otherwise. Since x_2 is good, the number of bad rows is at most $(c\rho(\mathsf{Bad}) + \varepsilon_2)R'$. However,

$$\begin{aligned}(c\rho(\mathsf{Bad}) + \varepsilon_2)R' \leq\; & cR'\sqrt{\varepsilon_1} + \varepsilon_2 R' \\ \leq\; & \frac{1}{2}(R')^{-0.6}R' + \varepsilon_2\left(\frac{1}{\varepsilon_2}\right)^{1+\alpha} <\; (R')^{0.4}.\end{aligned}$$

Next, fix t distinct *good* rows $y'_1, \ldots, y'_t$. Let $z_1, \ldots, z_t \in [A]$ be such that $S(x_2, y'_i, z_i) \notin \mathsf{Bad}$ (the z_i-s exist because we look at t good rows). Then, for every $i \in [t]$,

$$\Big(\mathsf{NM}(X_1, x_2; y'_i, z_i), \{\mathsf{NM}(X_1, x_2; y'_i, z)\}_{z \neq z_i}, \\ \{\mathsf{NM}(X_1, x_2; y'_j, z)\}_{j \neq i, z \in [A]}\Big)$$

is $\sqrt{\varepsilon_1}$-close to

$$\left(U_1, \{\mathsf{NM}(X_1, x_2; y'_i, z)\}_{z \neq z_i}, \{\mathsf{NM}(X_1, x_2; y'_j, z)\}_{j \neq i, z \in [A]}\right),$$

where we have used the fact that a good row is independent of any other tA (good or bad) rows (see Lemma 2.4.2), and the fact that S is simple. We remark that the property we are using is more than just the t-wise independence of the good rows.

By Lemma 2.1.2,

$$\Big(\mathsf{NM}(X_1, x_2; y'_1, z_1), \ldots, \mathsf{NM}(X_1, x_2; y'_t, z_t), \\ \{\mathsf{NM}(X_1, x_2; y'_i, z)\}_{(y'_i,z)\notin\{(y'_1,z_1),\ldots,(y'_t,z_t)\}}\Big)$$

is $t\sqrt{\varepsilon_1}$-close to

$$\left(U_t, \{\mathsf{NM}(X_1, x_2; y'_i, z)\}_{(y'_i,z)\notin\{(y'_1,z_1),\ldots,(y'_t,z_t)\}}\right).$$

This shows

$$(\oplus\mathsf{NM}(X_1, x_2, y'_1), \ldots, \oplus\mathsf{NM}(X_1, x_2, y'_t)) \approx_{t\sqrt{\varepsilon_1}} U_t$$

and $\oplus R(X_1, x_2)$ is a $(q = (R')^{0.4}, t, \gamma)$ non-oblivious bit-fixing source for $\gamma = t\sqrt{\varepsilon}_2$ as desired. □

Therefore, by Lemma 2.1.14, for any good x_2,

$$\left|\Pr[\mathrm{Maj}(\oplus\mathsf{NM}(X_1, x_2, 1), \ldots, \oplus\mathsf{NM}(X_1, x_2, R')) = 1] - \frac{1}{2}\right| \leq \\ c_{\mathsf{maj}}\left(\frac{\log t}{t} + (R')^{-0.1} + t\sqrt{\varepsilon}_2(R')^t\right) \leq 3 \cdot \frac{\varepsilon}{6} = \frac{\varepsilon}{2},$$

where the probability is over X_1. Overall, we have:

$$|\mathsf{2Ext}(X_1, X_2) - U_1| \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} \leq \varepsilon,$$

as desired.

The requested entropies are $k_1 = d$, $k_2 = O(d)$ and $d = O(f(n, \frac{1}{\mathsf{poly}(n)}))$ so

$$k_1, k_2 = O\left(\log n \cdot \frac{\log\log n}{\log\log\log n}\right).$$

The explicitness follows from the fact that $R' = \mathsf{poly}(n)$ and the explicitness of the other ingredients.

□

We note that if one improves the construction of non-malleable extractors and gets $f(n, \varepsilon_1) = O(\log \frac{n}{\varepsilon_1})$, then this would immediately imply a two-source extractors with entropies $k_1 = k_2 = O(\log n)$.

Chapter 4

Low-Error Two-Source Extractors from Good Non-Malleable Extractors

In spite of exciting improvements upon the min-entropy requirement of two-source extractors (some of which described in Chapter 3), explicit constructions of *low-error* two-source extractors still require one of the source to have min-entropy at least $0.49n$ [Bou05, Raz05]. Our main result in this chapter is a new *reduction* from explicit two-source extractors for min-entropy $n^{\Omega(1)}$ and negligible error to explicit t-non-malleable extractors with seed-length that has a good dependence on t.

Our reduction is inspired by the Chattopadhyay and Zuckerman framework [CZ16], however surprisingly we dispense with the use of resilient functions which are a major ingredient in [CZ16] and follow-up works. The use of resilient functions posed a fundamental barrier towards achieving negligible error, and our new reduction circumvents this bottleneck. The parameters we require from t-non-malleable extractors for our reduction to work hold in a non-explicit construction, but currently it is not known how to explicitly construct such extractors. As a result we do not give an unconditional construction of an explicit low-error two-source extractor. Nonetheless, we give a viable approach for solving the important problem of low-error two-source extractors. Furthermore, we will highlight an existing barrier in constructing low-error two-source extractors, and draw attention to the dependence of the parameter t in the seed-length of the non-malleable extractor.

4.1 Introduction

Recall that an $((n_1, k_1), (n_2, k_2), \varepsilon)$ two-source extractor is a function

$$\mathsf{E}\colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$$

that maps any pair of independent (n_1, k_1) and (n_2, k_2) sources X_1, X_2 to a distribution $\mathsf{E}(X_1, X_2)$ which is ε-close to U_m, the uniform distribution over $\{0,1\}^m$.

Non-explicitly there are $((n,k),(n,k),\varepsilon)$ two-source extractors as long as $k \geq \log n + 2\log(\frac{1}{\varepsilon}) + O(1)$. More generally, using the probabilistic method, one can easily show that:

Theorem 4.1.1. *Assume $k_1 + k_2 \geq \log(2^{k_1}n_1 + 2^{k_2}n_2) + 2\log(\frac{1}{\varepsilon}) + O(1)$. Then, there exists a (non-explicit) $((n_1, k_1), (n_2, k_2), \varepsilon)$ two-source extractor $\mathsf{E}\colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$.*

Finding such *explicit* constructions is a long-standing, important and challenging problem. A key parameter is the error ε obtained by the two-source extractor. Research in the area can be divided into three regimes:

Very large error: Finding explicit two-source extractors with any error smaller than 1 (i.e., any non-trivial error) is already very challenging and is essentially equivalent to finding an explicit *bipartite* Ramsey graph. Recall that a K *Ramsey graph* is a graph that contains no monochromatic set (i.e., a clique or an independent set) of size K; a K *bipartite Ramsey graph* is a bipartite graph with no bipartite monochromatic sets of size K. A $K = 2^k$ bipartite Ramsey graph over $2N = 2 \cdot 2^n$ vertices, is essentially equivalent to an $((n,k),(n,k),\varepsilon)$ two-source extractor, with $\varepsilon = \varepsilon(n) < 1$.

A long line of research was devoted to explicitly constructing Ramsey graphs (see Section 2.5.2 for an overview). Independently, Cohen [Coh16e] and Chattopadhyay and Zuckerman [CZ16] constructed a K bipartite Ramsey graph over $2N$ vertices with $\log K = \mathsf{polylog}(\log N)$. Improvements upon [CZ16], discussed in Chapter 3, readily give Ramsey graphs as well.

Medium size error: The [CZ16] construction of an explicit $((n,k),(n,k),\varepsilon)$ two-source extractor for $k = \mathsf{polylog}(n)$ has running time which is polynomial in $1/\varepsilon$. Several improvements followed, including [Mek17, Li16], and currently, following [BADTS17, Coh16f, Li17, Li18], the best explicit construction achieves k which is pretty close to the optimal $\Omega(\log n)$ bound.

All these constructions have running time which is at best polynomial in $1/\varepsilon$, and as we explain below this seems to be inherent to the approach that is taken. In contrast, non-explicit constructions may have exponentially small error in the entropy k of the two sources. Similarly, these constructions usually output few close-to-uniform bits, while non-explicitly, almost all of the entropy can be extracted.

Exponentially small error: There are several explicit two-source extractors constructions with exponentially small error:

1. The inner-product function gives a simple construction when $k > n/2$ [CG88].
2. Bourgain [Bou05] gave a two-source extractor construction for $k = \left(\frac{1}{2} - \alpha\right) n$, for some small constant $\alpha > 0$.
3. Raz [Raz05] constructed an $((n_1,k_1),(n_2,k_2),\varepsilon)$ two-source extractor that has an unbalanced entropy requirement; the first source is long (of length n_1) and very weak (k_1 can be as small as $\log\log n_1 + O(1)$), the second source is short (of length $O(\log n_1)$) and somewhat dense with $k_2 \geq \alpha n_2$, for some constant $\alpha > \frac{1}{2}$.

On the positive side, all of these constructions have exponentially small error (in Raz's extractor, the error is exponentially small in the smaller entropy). On the negative side, however, in all of these constructions one of the sources is required to have entropy rate close to half, i.e., the entropy of the source has to be at least $\left(\frac{1}{2} - \alpha\right) n > 0.49n$.

To summarize:

- Current explicit constructions of low-error, two-source extractors require one source to have entropy rate close to half, and,
- There are explicit two-source extractors that work with astonishingly low min-entropy, but currently they only handle large error, or, more precisely, their running time is polynomial in $1/\varepsilon$.

As we shall see shortly, there is a good reason for the two barriers that are represented in the above two items. The goal of this chapter is to present a new approach for bypassing these barriers.

4.1.1 Extractors and entropy-rate half

Let us start with the rate-half barrier for low-error constructions. For that we compare two-source extractors with *strong seeded extractors*. We recall that $\mathsf{E}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a strong (k,ε) seeded extractor if for every (n,k) source X, $(Y, \mathsf{E}(X,Y))$ is ε-close to $Y \times U_m$, where Y is uniformly distributed over $\{0,1\}^d$ and is independent of X.

A seeded extractor E must have seed-length $d \geq \log n + 2\log(\frac{1}{\varepsilon}) - O(1)$ [RTS00]. In essence, the error of a *seeded* extractor has two origins:

- The fraction ε_1 of bad seeds for which $\mathsf{E}(X,y)$ is ε_2-far from uniform, and,
- The distance ε_2 between $\mathsf{E}(X,y)$ and U_m for good seeds.

These two errors can be very different, for example, it might be the case that for half the seeds the error is extremely small, and then ε_1 is constant and ε_2 is tiny, or vice versa. In the terminology of a seeded extractor, these two errors are unified to one parameter ε. In the two-source extractor notation these two errors are essentially *separated*, where 2^{k_2} is, roughly, the number of bad seeds making $\varepsilon_1 \approx 2^{k_2-n_2}$, where ε of the two-source extractor represents the ε_2 above. More formally:

Claim 4.1.2. *Suppose* $\mathsf{E}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$ *is an* $((n,k),(d,d'),\varepsilon_2)$ *two-source extractor. Then,* E *is a strong* $(k, \varepsilon = \varepsilon_1 + \varepsilon_2)$ *seeded extractor, for* $\varepsilon_1 = 2^{d'+1-d}$*, and furthermore, for every* (n,k) *source* X*,*

$$\Pr_{y \in \{0,1\}^d}[\mathsf{E}(X,y) \not\approx_{\varepsilon_2} U_1] \;\leq\; \varepsilon_1.$$

Proof. Let X be an (n,k) source and let $B \subseteq \{0,1\}^d$ so that for every $y \in B$, $\mathsf{E}(X,y) \not\approx_{\varepsilon_2} U_1$. Partition $B = B_0 \cup B_1$ where $y \in B_z$ if the ε_2 bias is towards z. Assume towards contradiction that $|B_z| \geq 2^{d'}$ for some z and consider the flat distribution Y over the set B_z. Thus, $H_\infty(Y) \geq d'$ so $\mathsf{E}(X,Y) \approx_{\varepsilon_2} U_1$ but by our definition, $\mathsf{E}(X,Y)$ is biased towards z – a contradiction. Altogether, $|B| \leq 2^{d'+1}$ so $\varepsilon_1 \leq |B|/2^d = 2^{d'+1-d}$. □

The lower bound $d \geq \log n + 2\log(\frac{1}{\varepsilon}) - O(1)$ imposed on extractors, does not reveal which of the two errors forces d to be large. Stating it more precisely, define a $(k,\varepsilon_1,\varepsilon_2)$ function $\mathsf{E}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ so that for every (n,k) source X, $\Pr_{y \in \{0,1\}^d}[\mathsf{E}(X,y) \not\approx_{\varepsilon_2} U_1] \leq \varepsilon_1$. What is the dependence of d on ε_1 and ε_2?

The existence of $((n,k),(d=n,d'=O(\log n)),\varepsilon)$ two-source extractors, implies that the dependence of d on ε_1 might be very close $1\cdot\log(\frac{1}{\varepsilon_1})$. On the other hand, the dependence of d on ε_2 is larger, $d\geq d'\geq 2\log(\frac{1}{\varepsilon_2})$, since we can view E as a strong (d',ε_2) extractor $\{0,1\}^d\times\{0,1\}^k\to\{0,1\}$ and $d'\geq 2\log\frac{1}{\varepsilon_2}$ is again a lower bound [RTS00]. Thus, the two-source extractor terminology allows a finer characterization of the quality of an extractor, separating the two errors ε_1 and ε_2 above.

Looking at it that way we see why rate-half is a natural barrier: An extractor with seed-length dependence $2\log(\frac{1}{\varepsilon})$ guarantees that out of the $D=2^d$ possible seeds, at most $D^{\frac{1}{2}+\beta}$ are $D^{-\beta}$ bad. Thus, one can get an explicit two-source extractor, where the seed has some constant density $\frac{1}{2}+\beta$, and exponentially small error, by constructing an explicit strong seeded extractor with seed-length dependence $(2+\gamma)\log(\frac{1}{\varepsilon})$ for some small constant γ. Constructing a two-source extractor with d'/d below half necessarily means using techniques that do not apply to strong seeded extractors. Bourgain achieves that in an ingenious way, by using additive combinatorics together with the inner product function, but, at least so far, this approach can only handle min-entropies slightly below half.

4.1.2 The running time of the [CZ16] scheme

In Section 3.1.1 we described the [CZ16] two-source extractor construction and later gave the entropy-efficient reduction that allowed improvements upon their result. Here, we describe the bottleneck for achieving smaller error.

Recall that recent two-source extractor constructions builds upon two main ingredients: the existence of explicit non-malleable extractors and resilient functions. Let X_1 and X_2 be two independent (n,k) sources. The starting point is to use a t-non-malleable extractor nmExt with error ε_1 and seed-length d_1 to produce a table T_1 with $D_1=2^{d_1}$ entries, where the i-th entry is $\mathsf{nmExt}(X_1,i)$. Using the property of the non-malleable extractor, one can show that $1-\sqrt{\varepsilon_1}$ fraction of the rows are uniform and almost t-wise independent (in the sense that any t good rows are close to uniform). The remaining rows are, however, arbitrarily correlated with those rows. Then, they

- Use the second source X_2 to sample a sub-table T_2 with some D_2 rows of the table T_1, such that a fraction of at most ε_2 of its rows are bad, and every t good rows are $\sqrt{\varepsilon_1}$-close to uniform, and,
- Apply a resilient function $f\colon\{0,1\}^{D_2}\to\{0,1\}$ on the sub-table T_2. f has to be resilient against $\sqrt{\varepsilon_2}D_2$ bad players, and should perform correctly even when the good players are t-wise independent.

It turns out that the sub-table T_2 is $(D_2^t t\sqrt{\varepsilon_1})$-close to a table where the good players are *truly* t-wise independent (as required by f) and so it is enough to choose ε_1 small enough so that $D_2^t t\sqrt{\varepsilon_1}$ is small, and this proves the correctness of the construction.

While this beautiful approach does give an unbiased output bit, it seems that it is inherently bound to have running time polynomial in $1/\varepsilon$. This is because no matter which resilient function we use, even if there is just a single bad player among the D_2 players, then that player alone may have $1/D_2$ influence over the result (in fact, [KKL88] showed there

is a player with $\Omega(\frac{\log D_2}{D_2})$ influence) and therefore that player can bias the result by $1/D_2$. Thus, the running time, which is at least D_2, is at least $\Omega(\frac{1}{\varepsilon})$, and this is indeed a common feature of all the constructions so far that use that approach.

One could have hoped to sample a sub-table T_2 that w.h.p. avoids *all* bad players, thus dispensing with the use of the resilient function. This approach is futile: If T_2 avoids all bad players then every row y of it will do, so indeed $\mathsf{nmExt}(X, y)$ is close to uniform and we can compute it fast, allowing for a small error. However, this brings us back to the seeded extractors case, and we already saw this cannot handle densities above half.

4.1.3 Our result

The main result in this chapter is a reduction showing how to explicitly construct low-error two-source extractors given explicit t-non-malleable extractors with small seed-length dependence on t. Formally,

Theorem 4.1.3. *Suppose for some constant $\alpha > 0$, for every integers n_1, k_1, t and every $\varepsilon_1 > 0$ there exists an explicit function*

$$\mathsf{nmExt}\colon \{0,1\}^{n_1} \times \{0,1\}^{d} \to \{0,1\}^{m}$$

that is a (k_1, ε_1) t-non-malleable extractor with $d \leq \alpha t \cdot \log(\frac{1}{\varepsilon_1})$.
Then, there exists an explicit function

$$\mathsf{2Ext}\colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^{m}$$

that is a $((n_1, k_1), (n_2 = O(\frac{d}{\alpha}), k_2 = O(\alpha n_2)), 2\sqrt{\varepsilon_1})$ two-source extractor, where the constants hidden in the big-O notation are independent of α.

We first remark that such non-malleable extractors non-explicitly exist. In fact, much better parameters are possible:

Theorem 4.1.4. *Let n, k, t and ε be such that $k \geq (t+1)m + 2\log\frac{1}{\varepsilon} + \log d + 4\log t + 3$. Then, there exists a (k, ε) t-non-malleable extractor $\mathsf{nmExt} : \{0,1\}^{n} \times \{0,1\}^{d} \to \{0,1\}^{m}$ with $d \leq 2\log\frac{1}{\varepsilon} + \log(n-k) + 2\log(t+1) + 3$.*

The proof of the Theorem is based on [DW09], where they only handle the $t = 1$ case. The Theorem was also independently proved by Cohen and Shinkar [CS17]. For completeness we give the proof in Section 4.3.

So far, the main focus in explicit constructions of t-non-malleable constructions has been getting an optimal seed-length dependence on n and ε, and as we already mentioned, currently the best explicit constructions have seed-length (and min-entropy requirement) of $O(t^2(\log n + \log^{1+o(1)}\frac{1}{\varepsilon}))$ (see Section 2.4).

There is a huge gap between the dependence of the seed-length on t in the non-explicit construction of Theorem 4.1.4, where t contributes an *additive* $2\log t$ factor to the seed-length, and the explicit construction (say, Theorem 2.4.5) where t contributes a *multiplicative*

Dependence on t	k	n_2	k_2	
$\omega\left(t \log \frac{1}{\varepsilon}\right)$				The approach fails
$\alpha t \log(\frac{1}{\varepsilon})$	arbitrary	$O(\frac{d}{\alpha})$	$O(\alpha)n_2$	α is any constant
$t^\alpha \log(\frac{1}{\varepsilon})$ or better	arbitrary	$\mathsf{poly}_{\alpha,\beta}(d)$	n_2^β	For some constants $\alpha, \beta < 1$
$t^\alpha \log(\frac{1}{\varepsilon})$ or better	small enough	n	n^β	For some constant $\beta < 1$

Table 4.1: Bounds for $((n,k),(n_2,k_2),\varepsilon)$ two-source extractors assuming an explicit t n.m. extractor with various seed-length d dependence on t. In all cases, the error ε is low.

t^2 factor to the seed-length.[1] Correspondingly, the quality of the two-source extractors construction we give significantly improves with a better dependence of the seed-length on the parameter t. In Table 4.1 we list the two-source extractors constructions we get for:

- The current best explicit constructions (we get nothing),
- A quadratic improvement over currently best explicit (we improve upon Raz's extractor), and,
- A further polynomial improvement.

The parameters in the second row resemble those of Raz's extractor: one source is long with very low entropy, the other is short with constant entropy rate. The main difference is that in Raz's extractor the entropy rate has to be above half, whereas here, assuming the existence of the appropriate explicit non-malleable extractors, the entropy rate can be an arbitrarily small constant.

By allowing the seed-length of the non-malleable extractor to have an even better dependence on t (and non-explicitly it does), we succeed in supporting polynomially-small min-entropies. More specifically, if the seed-length dependence on t is $t^\alpha \log(\frac{1}{\varepsilon})$ for a small enough constant α, then we can support min-entropy of $k_2 = n_2^\beta$ where $\beta = \beta(\alpha)$ is another constant.

Also, in that regime of dependence, we can set the error ε to be small enough so that $n_2 = n$, in which case we get a *balanced* two-source extractor supporting some polynomially-small min-entropy (see Corollary 4.2.6).

This clearly demonstrates that the dependence of the seed-length on t in non-malleable extractors is directly related to the required density of the seed (i.e., second source) in low-error, two-source constructions. We believe this understanding is an important, qualitative understanding.

[1] It is worth mentioning that an early construction of Cohen, Raz and Segev [CRS14], although not explicitly stating it, does get a very good dependence of d on t with $d = O(\log \frac{n}{\varepsilon} + t)$. However, their construction only works for high min-entropy and so does not imply a two-source extractor for densities below half.

4.1.4 Our technique

In the Chattopadhyay-Zuckerman scheme we have the following ingredients:

1. The use of the first source to construct a table with many good rows (every row in the table corresponds to applying an extractor on the first source, with some fixed seed).
2. The use of t-non-malleable extractors to get *local* t-wise independence, where every t good rows are close to uniform.
3. The use of the second source to sample a sub-table of the table constructed from the first source.
4. The realization that with the right choice of parameters the sub-table is *globally* close to a table where the good rows are perfectly t-wise.
5. The use of resilient functions.

In our solution we keep (1)-(3) and completely dispense with (4) and (5), i.e., we do not use resilient functions and we do not try to achieve a sub-table that is *globally* close to a *truly* t-wise independent distribution. Instead, we work with the much weaker *local* guarantee that every t good rows are close to uniform.

Our construction is as follows. We are given two samples from independent sources $x_1 \sim X_1$ and $x_2 \sim X_2$. Then:

1. We use a t-non-malleable extractor nmExt with error ε_1 and seed-length d_1 to construct a table with $D_1 = 2^{d_1}$ entries, where the i-th entry is $\mathsf{nmExt}(X_1, i)$. Using the property of non-malleable extractors one can show that $1 - \sqrt{\varepsilon_1}$ fraction of the rows are good in the sense that a good row is close to uniform even conditioned on $t-1$ other rows. The remaining rows are arbitrarily correlated with the good ones. So far, everything is identical to the [CZ16] construction.
2. We use the second sample x_2 to sample t rows from that table, with the property that with high probability (over the choice of $x_2 \sim X_2$) **at least one of the t samples is a good row** (in the table with D_1 rows).

   We note that this is very different from previous constructions, where the requirement is that with high probability (over the choice of $x_2 \sim X_2$) the fraction of bad rows in the sub-table is about the same as the fraction of bad rows in the original table.
3. We then take the *parity* of the t strings written in the t rows we sampled.

   This is again very different from the [CZ16] construction, where a resilient function is applied on the sub-table (and notice that the parity function is not resilient at all).

Conceptually, what happened is that we take a *dramatically smaller* sample set than before. Specifically, in Chapter 3 the sample set is much larger than t, whereas in the above algorithm the sample size is t. Accordingly, we replace the requirement that the fraction of bad players in the sample set is small, with the weaker requirement that *not all* of the

players in the sample set are bad. If the sample size is t and not all the players in the sample are bad, then every good player (and even if there is just a single good player) is almost independent of the other $t-1$ players, and therefore we can just apply the parity function on the t bits in the sample. Thus, we can also dispense with the resilient function f and just use the parity function instead.

Notice that by doing so we also get rid of the annoying (and expensive) requirement that $D_2^t \varepsilon_1 < 1$, because we no longer need to convert a table where every t rows are locally close to uniform, to a table that is globally close to being perfectly t-wise independent. seed-length There is still a fundamental question we need to answer. Inspecting the argument, we see that there is a circular dependency in the construction: The sample size of the sampler determines the required t-non-malleability of the extractor, which then affects the parameters of the extractor, and in particular the number of bad rows, which, in turn, affects the required degree of the sampler. It is therefore, offhand, not clear whether such a construction is possible at all even assuming the best possible non-malleable extractors.

The above inquiry raises the question of what is the dependence of the seed-length of non-malleable extractors on the non-malleability parameter t. This question was considered before by several people. In particular, Cohen and Shinkar [CS17] independently investigated this. As we explained before, it turns out that in non-explicit constructions the dependence is very mild, and such an approach can be easily supported.

We analyze what is the threshold beyond which such an approach cannot work. Roughly speaking, non-malleable extractors with seed-length below $t \log(\frac{n}{\varepsilon})$ work well, while non-malleable extractors with seed-length above it do not. In Section 4.2 we demonstrate how the dependence of the seed-length d on t affects the parameters of the two-source extractor construction.

Finally, we are left with two questions regarding *explicitness*:

- We ask whether the sampler can be made explicit, i.e., whether we can find a sampler with such a small sample size that except for very few x_2-s always sees at least one good row. This question readily translates to the existence (or the explicit existence) of *dispersers* that are good against small tests. Remarkably, Zuckerman's dispersers from Section 2.3.1 work well for us.

- Current explicit constructions of non-malleable extractors [Coh16a, CGL16, Coh16c, Coh16b, CL16, Coh16f, Li17, Li18] for small entropies are above that threshold. This result raises the challenge of explicitly constructing non-malleable extractors with better seed-length dependence on t.

4.1.5 Related work

Li [Li12b] showed how to build a $((n, 0.499n), (n, k), 2^{-\Omega(n)})$ two-source extractor assuming a 1-non-malleable extractor with seed-length $d = 2\log(1/\varepsilon) + o(n)$. Li's work is orthogonal to ours. First, it asks for small seed dependence on the error: the seed-length of the non-malleable extractor has to be at most 2.001, while we look on the dependence on t. Also, it achieves limited parameters (even assuming non-explicit constructions) that are close to those in Bourgain's construction, and it is also close in spirit to Bourgain's construction.

We believe this result reveals an intrinsic connection between the dependence of the seed-length of a non-malleable extractor on the non-malleability parameter t and the quality of low-error two-source extractors, and is the first work to draw attention to the important problem of the dependence of the seed-length on t in explicit construction.

4.2 The Construction

4.2.1 The overall structure

Given:

$$\begin{aligned} \mathsf{nmExt} &: \{0,1\}^{n_1} \times [D] \to \{0,1\}^m \\ \Gamma &: \{0,1\}^{n_2} \times [t+1] \to [D] \end{aligned}$$

We define $\mathsf{2Ext}\colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ by

$$\mathsf{2Ext}(x_1, x_2) = \bigoplus_{y:\ \exists\ i \text{ s.t. } \Gamma(x_2,i)=y} \mathsf{nmExt}(x_1, y).$$

Theorem 4.2.1. *Assume* nmExt *is a* (k_1, ε_1) *t-non-malleable extractor and* Γ *is a* $(B_2, \sqrt{\varepsilon_1}D)$ *disperser. Then, for every* k_2, $\mathsf{2Ext}$ *is a* $\left((n_1,k_1),(n_2,k_2),\frac{B_2}{K_2}+\sqrt{\varepsilon_1}\right)$ *two-source extractor.*

Proof. Let X_1 be an (n_1, k_1) source and X_2 an (n_2, k_2) source. Without loss of generality X_1 and X_2 are flat. As nmExt is t-non-malleable, by Lemma 2.4.2 there exists a set $\mathsf{Bad}_1 \subseteq [D]$ with $\rho(BAD_1) \leq \sqrt{\varepsilon_1}$ such that for every $y \notin \mathsf{Bad}_1$ and every $y'_1, \ldots, y'_t \in [D] \setminus \{y\}$,

$$\left|\left(\mathsf{nmExt}(X,y), \{\mathsf{nmExt}(X,y'_i)\}_{i\in[t]}\right) - \left(U_m, \{\mathsf{nmExt}(X,y'_i)\}_{i\in[t]}\right)\right| \leq \sqrt{\varepsilon_1}.$$

Let $\mathsf{Bad}_2 \subseteq [N_2]$ be

$$\mathsf{Bad}_2 = \{x_2 \in \{0,1\}^{n_2} : \Gamma(x_2) \subseteq \mathsf{Bad}_1\}.$$

Thus, $\Gamma(\mathsf{Bad}_2) \subseteq \mathsf{Bad}_1$. Since $|\mathsf{Bad}_1| \leq \sqrt{\varepsilon_1}D$ and Γ_2 is a $(B_2, \sqrt{\varepsilon_1}D)$ disperser, it follows that $|\mathsf{Bad}_2| \leq B_2$. However, for any $x_2 \in \{0,1\}^{n_2} \setminus \mathsf{Bad}_2$, there exists an $i \in [t+1]$ such that $y = \Gamma(x_2, i) \notin \mathsf{Bad}_1$. Hence,

$$\left|\left(\mathsf{nmExt}(X,y), \{\mathsf{nmExt}(X,y_j)\}_{y_j\in\Gamma(x_2)\setminus\{y\}}\right) - \left(U_m, \{\mathsf{nmExt}(X,y_j)\}_{y_j\neq\Gamma(x_2)\setminus\{y\}}\right)\right| \leq \sqrt{\varepsilon_1}.$$

Thus,

$$\left|\bigoplus_{y:\ \exists\ i \text{ s.t. } \Gamma(x_2,i)=y} \mathsf{nmExt}(x_1,y) \ -\ U_m\right| \leq \sqrt{\varepsilon_1}.$$

Altogether, the error is at most $\frac{|\mathsf{Bad}_2|}{K_2} + \sqrt{\varepsilon_1}$ and the proof is complete. □

4.2.2 The activation threshold

In the previous subsection we assumed the existence of a $(B_2, \sqrt{\varepsilon_1}D)$ disperser Γ and a t-non-malleable extractor nmExt. However,

- The degree D_2 of the disperser Γ affects the non-malleability parameter t of the extractor, because the argument requires $t \geq D_2 - 1$,
- The non-malleability parameter t affects the degree $2^d = D$ of the extractor, because intuitively, the greater t is the greater the degree has to be,
- The degree D determines $|\mathsf{Bad}_1| = \sqrt{\varepsilon_1}D$, and,
- The size B_1 of the set Bad_1 determines the degree of the disperser Γ as $D_2 = O\left(\frac{\log \frac{N_2}{B_2}}{\log \frac{D}{B_1}}\right)$, and up to a multiplicative factor this is also a lower bound on D_2.

Thus we have a circular dependence and it is not clear at all that such a construction is even possible. Indeed, as we shall see, if the seed-length of nmExt is larger than $t\log(\frac{1}{\varepsilon_1})$ such a construction is impossible. However, at least non-explicitly, better non-malleable extractors exist that comfortably suffice for the construction. Our goal in this section is to determine which dependence of the seed-length on t and ε_1 suffices for the construction.

4.2.3 The analysis fails when $d \geq ct\log(\frac{1}{\varepsilon})$ for some constant c

Lemma 4.2.2. *Suppose*

$$\begin{aligned} \mathsf{nmExt} &: \{0,1\}^{n_1} \times [D] \to \{0,1\}^m \\ \Gamma &: \{0,1\}^{n_2} \times [t+1] \to [D] \end{aligned}$$

are such that nmExt *is a* (k_1, ε_1) *t-non-malleable extractor and* Γ *is any* $(B_2, B_1 = \sqrt{\varepsilon_1}D)$ *disperser, as required by Theorem 4.2.1. Suppose Theorem 4.2.1 gives that*

$$\mathsf{2Ext}\colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$$

is an $((n_1,k_1),(n_2,k_2),2\sqrt{\varepsilon_1})$ *two-source extractor with* $K_2 < \sqrt{N_2}$*. Then,* $\log_{1/\varepsilon_1} D \leq \frac{t+1}{c_0}$*, where* c_0 *is the constant guaranteed by Theorem 2.3.3.*

Proof. We first give some easy bounds on the parameters:

- $B_2 \leq K_2$, for otherwise Theorem 2.3.4 constructs $\mathsf{2Ext}$ with the trivial error 1.
- Also, $tB_2 \geq B_1$, for otherwise we can take a set $A \subseteq \{0,1\}^{n_2}$ of cardinality B_2 and the size of its neighbor set is at most $B_2 t < B_1$ violating the disperser property.
- Finally, $\frac{B_1}{t} \geq \sqrt{B_1}$ because otherwise $\sqrt{B_1} < t$ and then

$$D_1 \;=\; \frac{B_1}{\sqrt{\varepsilon_1}} \;<\; \frac{t^2}{\sqrt{\varepsilon_1}} \leq \frac{n_1^2}{\sqrt{\varepsilon_1}} \;\leq\; \frac{1}{\varepsilon_1^2},$$

  where the last inequality follows from the assumption on ε_1. This contradicts the lower-bound for extractors [RTS00].

Together,

$$\frac{N_2}{B_2} \geq \frac{N_2}{K_2} \geq K_2 \geq B_2 \geq \frac{B_1}{t} \geq \sqrt{B_1} = \sqrt{\varepsilon_1}D,$$

and $\frac{D}{B_1} = \frac{1}{\sqrt{\varepsilon_1}}$. Now, $\Gamma : \{0,1\}^{n_2} \times [t+1] \to [D]$ is a $(B_2, B_1 = \sqrt{\varepsilon_1}D)$ disperser and therefore by Theorem 2.3.3 it has degree at least $c_0 \cdot \frac{\log \frac{N_2}{B_2}}{\log \frac{D}{B_1}}$ for some constant c_0. Therefore,

$$\begin{aligned} t+1 &\geq c_0 \cdot \frac{\log \frac{N_2}{B_2}}{\log \frac{D}{B_1}} \geq c_0 \cdot \frac{\log \sqrt{\varepsilon_1}D}{\log \frac{1}{\sqrt{\varepsilon_1}}} \\ &= 2c_0 \cdot \log_{1/\varepsilon_1}(\sqrt{\varepsilon_1}D) = 2c_0 \cdot (\log_{1/\varepsilon_1} D - 1/2) \geq c_0 \log_{1/\varepsilon_1} D. \end{aligned}$$

□

The analysis in the above proof is quite tight and in the next subsection we prove the converse (which also entails Theorem 4.1.3).

4.2.4 When $d = O(t \log(\frac{1}{\varepsilon}))$

Lemma 4.2.3. *Let $\varepsilon_1 \leq \frac{1}{n}$. Suppose there exists an explicit*

$$\mathsf{nmExt}\colon \{0,1\}^{n_1} \times [D_1] \to \{0,1\}^m$$

that is a (k_1, ε_1) t-non-malleable extractor with $\log_{1/\varepsilon_1} D_1 \leq \frac{\alpha}{8c_{\mathsf{Disp}}} t$ for some constant $\alpha > 0$, some constant t and some k_1. Then, there exists an explicit function

$$\mathsf{2Ext}\colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$$

that is a $((n_1, k_1), (n_2 = \frac{4}{\alpha}d_1, k_2 = \alpha n_2), 2\sqrt{\varepsilon_1})$ two-source extractor.

Proof. Fix t as in the hypothesis of the lemma. Set D such that $\log_{1/\varepsilon_1} D = \frac{\alpha t}{8c_{\mathsf{Disp}}}$. Let

$$\Gamma\colon [N_2 = D^{4/\alpha}] \times [D_2] \to [D]$$

be the $(B_2 = D^2, B_1 = \sqrt{\varepsilon_1}D)$ disperser promised to us by Theorem 2.3.4 for $a = \frac{\alpha}{2}$ (because $B_2 = N_2^a$) and $b = \frac{1}{2}$ (because $D = B_2^b$). By Theorem 2.3.4 the degree D_2 of Γ is

$$\begin{aligned} D_2 &= c_{\mathsf{Disp}} \cdot \frac{\log \frac{N_2}{B_2}}{\log \frac{D}{B_1}} = c_{\mathsf{Disp}} \cdot \frac{\frac{4(1-a)}{\alpha} \log D}{\log \frac{1}{\sqrt{\varepsilon_1}}} \\ &= c_{\mathsf{Disp}} \cdot \left(\frac{1}{\alpha} - \frac{1}{2}\right) \frac{8 \log D}{\log 1/\varepsilon_1} = c_{\mathsf{Disp}} \cdot \left(\frac{8}{\alpha} - 4\right) \log_{1/\varepsilon_1} D \\ &= c_{\mathsf{Disp}} \cdot \left(\frac{8}{\alpha} - 4\right) \frac{\alpha t}{8c_{\mathsf{Disp}}} = \left(1 - \frac{\alpha}{2}\right) t < t. \end{aligned}$$

Let

$$\mathsf{nmExt}\colon \{0,1\}^{n_1} \times [D_1] \to \{0,1\}^m$$

be the explicit (k_1, ε_1) t-non-malleable extractor with $\log_{1/\varepsilon_1} D_1 \leq \frac{\alpha}{8c_{\mathsf{Disp}}} t = \log_{1/\varepsilon_1} D$ promised by the hypothesis of the lemma. As $\frac{1}{\varepsilon} > 1$, we see that $D_1 \leq D$ and we may take D_1 larger so that it equals D.

Now, let

$$\mathsf{2Ext}\colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$$

be constructed from nmExt and Γ as above. As nmExt is a (k_1, ε_1) t-non-malleable extractor and Γ is a $(B_2, \sqrt{\varepsilon_1} D_1)$ disperser, Theorem 4.2.1 tells us that for every k_2, $\mathsf{2Ext}$ is a $((n_1, k_1), (n_2, k_2), \frac{B_2}{K_2} + \sqrt{\varepsilon_1})$ two-source extractor. Taking $k_2 = \alpha n_2$,

$$\frac{B_2}{K_2} + \sqrt{\varepsilon_1} = \frac{D_1^2}{D_1^4} + \sqrt{\varepsilon_1} = \frac{1}{D_1^2} + \sqrt{\varepsilon_1}.$$

But $D_1 \geq \frac{1}{\varepsilon_1^2}$ (this is true for any seeded extractor [RTS00]). Altogether the error is at most $\sqrt{\varepsilon_1} + \frac{1}{\varepsilon_1^2} \leq 2\sqrt{\varepsilon_1}$. □

4.2.5 When $d = O(t^\alpha \log \frac{1}{\varepsilon})$

A careful examination of the parameters shows that if the dependence of d_1 on t is better, our scheme yields a two-source extractor that supports even smaller min-entropies. Roughly speaking, if $\log_{1/\varepsilon_1} D_1 = t^\alpha$ for some $\alpha < 1$ we can support some polynomially-small min-entropy $k_2 = n_2^\beta$, instead of only supporting min-entropies of constant rate. Specifically:

Lemma 4.2.4. *Let $\varepsilon_1 \leq \frac{1}{n}$. There exists a constant $\beta_0 < 1$ such that for every $\beta_0 < \beta < 1$ there exist constants $\alpha < 1$ and $\gamma > 1$ so that the following holds. Suppose there exists an explicit*

$$\mathsf{nmExt}\colon \{0,1\}^{n_1} \times [D_1] \to \{0,1\}^m$$

that is a (k, ε_1) t-non-malleable extractor with $\log_{1/\varepsilon_1} D_1 \leq t^\alpha$ for some k_1, and t which is a large enough polynomial in $\log \frac{1}{\varepsilon_1}$. Then there exists an explicit

$$\mathsf{2Ext}\colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$$

that is a $((n_1, k_1), (n_2 = d_1^\gamma, k_2 = n_2^\beta), 2\sqrt{\varepsilon_1})$ two-source extractor.

The proof is similar to the proof of Lemma 4.2.3. However, it is no longer true that K_2 is a *constant* power of N_2, so we should be more careful with the parameters of Zuckerman's disperser. Particularly, in this regime of parameters, the degree D_2 (and consequently t) is no longer constant but will be poly-logarithmic in $\frac{1}{\varepsilon}$. The following Theorem extends Theorem 2.3.4 for the more general case.

Theorem 4.2.5 ([Zuc07])**.** *There exist constants $c_1, c_2 > 1$ such that the following holds. For every $0 < \delta < 1$, N, $K = N^\delta$, $M \leq N^{\delta^{c_2}}$ and $K' < M$ there exists an efficient family of (K, K') dispersers*

$$\Gamma\colon [N] \times [D] \to [M]$$

with degree $D = \left(\frac{1}{\delta}\right)^{c_1} \cdot \frac{n}{\log \frac{M}{K'}}$.

We are now ready to prove Lemma 4.2.4.

Proof. Let c_1 and c_2 be as in Theorem 4.2.5. Set $\beta_0 = 1 - \frac{1}{c_2}$ and fix some $\beta_0 < \beta < 1$. Fix t as in the hypothesis of the lemma. Set D such that $\log_{1/\varepsilon_1} D = t^\alpha$ for $\alpha = \alpha(\beta)$ we will soon explicitly determine. Let

$$\Gamma \colon [N_2 = D^{1/\delta c_2}] \times [D_2] \to [D]$$

be the $(B_2 = N_2^\delta, B_1 = \sqrt{\varepsilon_1} D)$ disperser promised to us by Theorem 4.2.5, for $\delta = \frac{1}{2} n_2^{-(1-\beta)}$. Notice that $b_2 = \delta n_2 = \frac{1}{2} n_2^\beta$ and set $k_2 = 2b_2 = n_2^\beta$. Also, observe that $n_2 = \frac{1}{\delta^{c_2}} d = (2^{c_2} d)^{\gamma'}$ for

$$\gamma' \;=\; \frac{1}{1 - c_2(1-\beta)}.$$

As $\beta > \beta_0$ we see that $\gamma' > 1$. It follows that $n_2 = d^\gamma$ for some $\gamma' < \gamma < 2\gamma'$.

By Theorem 4.2.5, the degree D_2 of Γ is

$$\begin{aligned} D_2 &= \left(\frac{1}{\delta}\right)^{c_1} \cdot \frac{n_2}{\log \frac{D}{B_1}} = \left(\frac{1}{\delta}\right)^{c_1} \cdot \frac{2n_2}{\log(1/\varepsilon_1)} \\ &= \left(2n_2^{1-\beta}\right)^{c_1} \cdot \frac{2 \cdot n_2}{\log(1/\varepsilon_1)} = 2^{c_1+1} \cdot \frac{n_2^{1+c_1(1-\beta)}}{\log(1/\varepsilon_1)} = 2^{c_1+1} \cdot \frac{(\log D)^{\gamma(1+c_1(1-\beta))}}{\log(1/\varepsilon_1)}. \end{aligned}$$

Set $\xi = \gamma(1 + c_1(1-\beta)) > 1$ and $\alpha = \frac{1}{2\xi}$ (note that α is in fact a function of β). We get that:

$$D_2 = 2^{c_1+1} \frac{\log^\xi D}{\log(1/\varepsilon_1)} = 2^{c_1+1} \left(\log^{\xi-1} \frac{1}{\varepsilon_1}\right) \left(\log_{1/\varepsilon_1} D\right)^\xi = 2^{c_1+1} \left(\log^{\xi-1} \frac{1}{\varepsilon_1}\right) t^{\alpha\xi}.$$

Now, note that $t^{\alpha\xi} = \sqrt{t}$, so $D_2 < t$ as long as $t > 4^{c_1+1} \log^{2(\xi-1)} \frac{1}{\varepsilon_1}$.

Let

$$\mathsf{nmExt} \colon \{0,1\}^{n_1} \times [D_1] \to \{0,1\}^m$$

be the explicit (k_1, ε_1) t-non-malleable extractor with $\log_{1/\varepsilon_1} D_1 \le t^\alpha = \log_{1/\varepsilon_1} D$ promised by the hypothesis of the lemma. Again, we can take $D_1 = D$.

Now let

$$\mathsf{2Ext} \colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$$

be constructed from nmExt and Γ as in Section 4.2.1. We have that nmExt is a (k_1, ε_1) t-non-malleable extractor and Γ is a $(B_2, \sqrt{\varepsilon_1} D_1)$ disperser, so by Theorem 4.2.1, $\mathsf{2Ext}$ is a $((n_1, k_1), (n_2, k_2), \frac{B_2}{K_2} + \sqrt{\varepsilon_1})$ two-source extractor.

In our case, $\frac{B_2}{K_2} = 2^{b_2 - k_2} = 2^{-b_2}$. We stress that $b_2 \ge \frac{1}{2} \log(\frac{1}{\varepsilon_1})$. To see this, note that $2b_2 = n_2^\beta = d_1^{\beta\gamma}$. As $\beta\gamma \ge \beta\gamma' = \frac{\beta}{1 - c_2(1-\beta)} \ge 1$, and $d_1 \ge 2 \log(\frac{1}{\varepsilon_1})$ (again, this is true for any seeded extractor), we finally have that $2b_2 \ge d_1 > \log(\frac{1}{\varepsilon_1})$. Overall,

$$\frac{B_2}{K_2} + \sqrt{\varepsilon_1} \;\le\; 2\sqrt{\varepsilon_1}$$

and we are done. □

Next, we show that we can *balance* the above two-source extractor (i.e., $n_1 = n_2$) by choosing the error ε_1 appropriately and assuming k_1 is small enough. The resulting two-source extractor supports polynomially-small min-entropies from both sources. Formally:

Corollary 4.2.6. *Let $\varepsilon_1 \leq \frac{1}{n}$. There exists a constant $\beta_0 < 1$ such that for every $\beta_0 < \beta < 1$ there exits a constant $\alpha < 1$ so that the following holds. Suppose there exists an explicit*

$$\mathsf{nmExt}\colon \{0,1\}^{n_1} \times [D_1] \to \{0,1\}^m$$

that is a (k_1, ε_1) t-non-malleable extractor with $\log_{1/\varepsilon_1} D_1 \leq t^\alpha$ for some $k_1 \leq d_1$, and t which is a large enough polynomial in $\log(\frac{1}{\varepsilon_1})$. Then there exists an explicit

$$\mathsf{2Ext}\colon \{0,1\}^{n} \times \{0,1\}^{n} \to \{0,1\}^m$$

that is an $((n, k = k_1), (n, k), \varepsilon)$ two-source extractor for $k = n^\beta$ and $\varepsilon = 2^{-n^{\Omega(1)}}$.

Proof. Following the notations of Lemma 4.2.4, let β_0, α, γ be the constants set according to β. Let $\mathsf{2Ext} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ be the explicit $((n_1, k_1), (n_2 = d_1^\gamma, k_2 = n_2^\beta), 2\sqrt{\varepsilon_1})$ two-source extractor that is guaranteed to us.

We require $n = n_1 = n_2 = d_1^\gamma$, so as $d_1 = t^\alpha \log(\frac{1}{\varepsilon_1}) \leq t \log(\frac{1}{\varepsilon_1})$ and t is polynomial in $\log(\frac{1}{\varepsilon_1})$, denote $t \log \frac{1}{\varepsilon_1} = \log^{\eta'} \frac{1}{\varepsilon_1}$ and $n = \log^\eta \frac{1}{\varepsilon_1}$ for some large enough constants $\eta', \eta = \gamma\eta'$. This guarantees that $\varepsilon = 2\sqrt{\varepsilon_1} = 2^{-n^{\Omega(1)}}$.

Next, note that $k_1 \leq d_1$ and $d_1 = n^{\frac{1}{\gamma}}$. Indeed, $n^{\frac{1}{\gamma}} \leq n^\beta$ since we already observed in the proof of Lemma 4.2.4 that $\gamma\beta \geq 1$. Overall $k_1 \leq n^\beta$ for every $\beta > \beta_0$. As by construction $k_2 = n_2^\beta$ for every $\beta > \beta_0$ as well, the proof is concluded. □

4.3 The Seed's Dependence on the Non-Malleability

In this section we extend the [DW09] result, where non-malleability was considered only in the case of $t = 1$. We repeat Theorem 4.1.4 and prove:

Theorem 4.3.1. *Let n, k, t and ε be such that $k \geq (t+1)m + 2\log\frac{1}{\varepsilon} + \log d + 4\log t + 3$. There exist a (k, ε) t-non-malleable extractor $\mathsf{nmExt}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d \leq 2\log\frac{1}{\varepsilon} + \log(n-k) + 2\log(t+1) + 3$.*

This was also independently proved by Cohen and Shinkar [CS17].

Proof. Choose a function $\mathsf{nmExt}\colon [N] \times [D] \to [M]$ uniformly at random. Fix a flat source X (which we identify with a subset $X \subseteq [N]$ of size K), t functions $f_1, \ldots, f_t : [D] \to [D]$ with no fixed-points and a distinguisher function $\mathcal{D}\colon \{0,1\}^{(t+1)m+d} \to \{0,1\}$. We want to bound the probability (over nmExt) that

$$\begin{aligned}\Pr[\mathcal{D}(\mathsf{nmExt}(X,Y), \mathsf{nmExt}(X, f_1(Y)), \ldots, E(\mathsf{nmExt}, f_t(Y)), Y) = 1] - \\ \Pr[\mathcal{D}(U_m, \mathsf{nmExt}(X, f_1(Y)), \ldots, \mathsf{nmExt}(X, f_t(Y)), Y) = 1] \ > \ \varepsilon.\end{aligned}$$

For every $y \in [D]$ and $z_1, \ldots, z_t \in [M]$, define

$$\mathrm{Count}(y, z_1, \ldots, z_t) \;=\; \left|\{z \in [M] : \mathcal{D}(z, z_1, \ldots, z_t, y) = 1\}\right|.$$

For every $x \in X$ and $y \in [D]$, define the following random variables (where the randomness comes from nmExt):

$$\begin{aligned}
\mathbf{L}(x,y) &= \mathcal{D}(\mathsf{nmExt}(x,y), \mathsf{nmExt}(x, f_1(y)), \ldots, \mathsf{nmExt}(x, f_t(y)), y) \\
\mathbf{R}(x,y) &= \frac{1}{M} \cdot \mathrm{Count}(y, \mathsf{nmExt}(x, f_1(y)), \ldots, \mathsf{nmExt}(x, f_t(y))) \\
\mathbf{Q}(x,y) &= \mathbf{L}(x,y) - \mathbf{R}(x,y) \\
\overline{\mathbf{Q}} &= \frac{1}{KD} \sum_{x \in X, y \in [D]} \mathbf{Q}(x,y).
\end{aligned}$$

As we mentioned above, we want to bound $\Pr[\overline{\mathbf{Q}} > \varepsilon]$. Notice that for every $x \in X$ and $y \in [D]$, due to the fact that $f_1, \ldots, f_t$ have no fixed points, we have that $\mathbb{E}[\mathbf{L}(x,y)] = \mathbb{E}[\mathbf{R}(x,y)]$ and thus $\mathbb{E}[\overline{\mathbf{Q}}] = 0$. However, the values of $\mathbf{Q}$ on different inputs are not independent.

To see why the $\mathbf{Q}$-s are not independent, think for example about the case where $t = 2$ and y is such that $f_2(f_1(y)) = y$. In such a scenario,

$$\begin{aligned}
\mathbf{L}(x,y) &= \mathcal{D}(\mathsf{nmExt}(x,y), \mathsf{nmExt}(x, f_1(y)), \mathsf{nmExt}(x, f_2(y)), y) \\
\mathbf{L}(x, f_1(y)) &= \mathcal{D}(\mathsf{nmExt}(x, f_1(y)), \mathsf{nmExt}(x, f_1(f_1(y))), \mathsf{nmExt}(x,y), f_1(y)),
\end{aligned}$$

so, depending on $\mathcal{D}$, $\mathbf{Q}(x,y)$ and $\mathbf{Q}(x, f_1(y))$ may not be independent. Luckily, it is sufficient to disregard such cycles in order to obtain sufficient "independence".

Let $G = (V = [D], E)$ be a directed graph (multiple edges allowed) such that

$$E \;=\; \{(y, f_k(y)) : y \in [D], k \in [t]\},$$

so the out-degree of every vertex is exactly t.

Lemma 4.3.2. *Assume that there exists a subset $V' \subseteq V$ such that the induced subgraph $G' \subseteq G$ is acyclic. Then, the set $\{\mathbf{Q}(x,y)\}_{x \in X, y \in V'}$ can be enumerated by $\mathbf{Q}_1, \ldots, \mathbf{Q}_{m = K|V'|}$ such that*

$$\mathbb{E}[\mathbf{Q}_i \mid \mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1}] \;=\; 0$$

for every $i \in [m]$.

Proof. G' is acyclic so it induces a partial order on V'. Use this partial order to induce a total order on $\{1, \ldots, m\}$ such that if $(y, y') \in E$ and $\mathbf{Q}_j = \mathbf{Q}(x, y')$, $\mathbf{Q}_i = \mathbf{Q}(x,y)$ then $j \leq i$.

Fix some $i \in [m]$ and assume $\mathbf{Q}_i = \mathbf{Q}(x,y)$. The key point is that the variables $\mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1}$ never query nmExt on the input (x,y). Conditioned on any choice of the value of nmExt for all points other than (x,y), denote them by $e_1, \ldots, e_t$, we have that

$$\mathbb{E}[\mathbf{Q}_i] = \mathbb{E}\left[\mathcal{D}(\mathsf{nmExt}(x,y), e_1, \ldots, e_t, y) - \frac{1}{M} \cdot \mathrm{Count}(y, e_1, \ldots, e_t)\right] = 0,$$

and as we noted, $\mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1}$ are deterministic functions of nmExt and independent of $\mathsf{nmExt}(x,y)$. □

We now need a partition of the vertices of G into acyclic induced subgraphs. The following lemma shows that such a partition exists with a small number of sets.

Lemma 4.3.3 ([NL82])**.** *For any directed graph $G = (V, E)$ with maximum out-degree t (multiple edges allowed), there exists a partition $V = V_1 \cup \ldots \cup V_{t+1}$ such that for every $i \in [t+1]$, the subgraph of G induced by V_i is acyclic.*

In light of the above two lemmas, there exists a partition of $\{\mathbf{Q}(x,y)\}_{x\in X, y\in[D]}$ to $t+1$ sets $\left\{\mathbf{Q}_1^1,\ldots,\mathbf{Q}_{s_1}^1\right\},\ldots,\left\{\mathbf{Q}_1^t,\ldots,\mathbf{Q}_{s_t}^t\right\}$ such that for every $k \in [t+1]$ and $i \in [s_k]$, $\mathbb{E}[\mathbf{Q}_i^k \mid \mathbf{Q}_1^k,\ldots,\mathbf{Q}_{i-1}^k] = 0$. Now, define $S_i^k = \sum_{j=1}^{i} \mathbf{Q}_j^k$ and note that every sequence $S_1^k,\ldots,S_{s_k}^k$ is a martingale. Also, $|S_i^k - S_{i-1}^k| = |\mathbf{Q}_i^k| \le 1$ with probability 1. Thus, using Azuma's inequality,

$$\begin{aligned}\Pr[\overline{\mathbf{Q}} > \varepsilon] = \Pr\left[\sum_{k=1}^{t+1} S_{s_k}^k > \varepsilon KD\right] &\le \sum_{k=1}^{t+1} \Pr\left[S_{s_k}^k > \frac{\varepsilon KD}{t+1}\right] \\ &\le \sum_{k=1}^{t+1} \exp\left(-\frac{\left(\frac{\varepsilon KD}{t+1}\right)^2}{2\cdot s_k}\right) \le (t+1)e^{-\frac{\varepsilon^2 KD}{2(t+1)^2}},\end{aligned}$$

where the last inequality follows from the fact that $s_k \le KD$.

To complete our analysis, we require E to work for *any* X, $f_1,\ldots,f_t$ and $\mathcal{D}$. By the union bound, the probability for a random E to fail, denote it by p_E, is given by

$$\begin{aligned}p_E &\le \binom{N}{K} D^{tD} 2^{D\cdot M^{t+1}} (t+1) e^{-\frac{\varepsilon^2 KD}{2(t+1)^2}} \\ &\le 2^{K\log\left(\frac{Ne}{K}\right) + tDd + DM^{t+1} + \log(t+1) - \frac{\varepsilon^2 KD\log e}{2(t+1)^2}} \\ &\le 2^{K(n-k+2) + tDd + DM^{t+1} + \log(t+1) - \frac{\varepsilon^2 KD}{2(t+1)^2}}.\end{aligned}$$

To prove that $p_E < 1$ (in fact this will show $p_E \ll 1$) it is sufficient to prove that:

1. $K(n-k+2) \le \frac{\varepsilon^2 KD}{8(t+1)^2}$.
2. $D(td + M^{t+1}) + \log(t+1) \le \frac{\varepsilon^2 KD}{8(t+1)^2}$, or alternatively $D(2td + M^{t+1}) \le \frac{\varepsilon^2 KD}{8(t+1)^2}$.

Item (1) is true whenever

$$D \;\ge\; \frac{8(t+1)^2(n-k+2)}{\varepsilon^2}.$$

Item (2) is true whenever

$$K \;\ge\; \frac{8(t+1)^2(2td + M^{t+1})}{\varepsilon^2}.$$

The bounds on d and k follow from the above two inequalities. □

Chapter 5

Low-Error Two-Source Condensers

As we discussed in previous chapters, the Chattopadhyay and Zuckerman two-source extractors for n-bit sources and error ε supporting low min-entropy run in time $\mathsf{poly}(n/\varepsilon)$. This means that with polynomial-time constructions, only polynomially-small errors are possible. Constructions that followed, supporting even lower entropies, did not improve upon the error guarantee and indeed in Chapter 4 we examined the low-error problem and gave a *conditional* solution.

Our main result in this chapter is an unconditional, $\mathsf{poly}(n, \log(1/\varepsilon))$-time computable two-source *condenser*. For any $k \geq \mathsf{polylog}(n/\varepsilon)$, our condenser transforms two independent (n, k) sources to a distribution over $m = k - O(\log(1/\varepsilon))$ bits that is ε-close to having min-entropy $m - o(\log(1/\varepsilon))$. Hence, achieving entropy gap of $o(\log(1/\varepsilon))$.

Recall that the bottleneck for obtaining low error in recent constructions of two-source extractors lies in the use of resilient functions: The error of a resilient function that receives input bits from r players cannot be smaller than $1/r$. This, in return, forces the running time of the construction to be polynomial in $1/\varepsilon$.

A key component in our construction in this chapter is a variant of resilient functions which we call *entropy-resilient functions*. This variant can be seen as playing the above game for several rounds. The goal of the corrupted players is to reduce, with as high probability as they can, the min-entropy accumulated throughout the rounds. We show that while the bias decreases only polynomially with the number of players in a one-round game, their success probability decreases *exponentially* in the entropy gap they are attempting to incur in a repeated game.

5.1 Introduction

5.1.1 Entropy-resilient functions

As discussed in Section 4.1.2, the [CZ16] extractor and subsequent constructions are not *strongly* polynomial-time, and, in particular, the error guarantee cannot be taken to be sub-polynomial in n while maintaining $\mathsf{poly}(n)$ running-time. In Chapter 4, we showed that if certain t-non-malleable extractors can be explicitly constructed then resilient functions can be replaced by the parity function (which is not resilient at all) and low-error two-source

extractors with low min-entropy requirement can be obtained. However, it is not known how to explicitly construct such t-non-malleable extractors.

To obtain our condenser, we extend the notion of resilient functions to functions outputting many bits. Informally speaking, instead of considering an r-player game in which the bad players try to bias the output, we study a repeated game version in which the r players play for m rounds. The bad players attempt to decrease, with as high probability as they can, the min-entropy of the m-bit outcome (and we will even allow the bad players to cast their votes after the good players played all rounds).

Recall that, by [KKL88], when $m = 1$, even a single player can bias the result by $\Omega(\frac{\log r}{r})$. Put differently, viewing this bias as the error of a deterministic extractor, the error is bound to be at least polynomially-small in the number of players. Our key insight is that when m becomes large, the probability that the bad players can reduce g bits of entropy from the output (creating an "entropy gap" of g) is *exponentially small* in g. We further show that this holds for a specific function f, induced by the Ajtai-Linial function, even when the honest players are only t-wise independent (for $t = \mathsf{polylog}(r/\varepsilon)$). Our analysis uses and extends ideas from [CZ16].

5.1.2 The two-source condensers we obtain

The main result in this chapter is an explicit construction of a *two-source condenser* with low error and small entropy gap, outputting almost all of the entropy from one source.

Definition 5.1.1. *A function* $\mathsf{Cond}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *is a* two-source condenser for min-entropy k with min-entropy gap g and error guarantee ε *if for every pair of independent* (n,k) *sources,* $\mathsf{Cond}(X,Y)$ *is* ε*-close to an* $(m, m-g)$*-source.* ◊

Note that a two-source extractor is a two-source condenser with entropy gap $g = 0$. Thus, condensers can be seen as a relaxation of extractors in which some, hopefully small, "gap" of min-entropy in the output distribution is allowed.

Despite having a weaker guarantee, condensers play a key role in the construction of many types of randomness extractors. We mentioned the role of seeded condensers in constructing seeded extractors (see Section 2.3.2), but the same is also true for multi-source condensers and multi-source extractors (see Chapter 3 and also [BKS+10, Zuc07, Rao09a, Li13a]). Most related to our work is a paper by Rao [Rao08] that, for every $\delta > 0$, constructed a $\mathsf{poly}(n, \log(1/\varepsilon))$-time computable two-source condenser[1] for min-entropy $k = \delta n$ having $m = \Omega(\delta n)$ output bits with entropy gap $g = \mathsf{poly}(1/\delta, \log(1/\varepsilon))$.

In this chapter, we obtain a strongly polynomial-time construction of a two-source condenser with low error and small min-entropy gap.

Theorem 5.1.2. *For all integers* n, k *and every* $\varepsilon > 0$ *such that* $n \geq k \geq \mathsf{polylog}(\frac{n}{\varepsilon})$*, there exists a* $\mathsf{poly}(n, \log(1/\varepsilon))$*-time computable two-source condenser*

$$\mathsf{Cond}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$$

for min-entropy k*, with error guarantee* ε*, min-entropy gap* $g = o(\log \frac{1}{\varepsilon})$*, and* $m = k - O(\log(1/\varepsilon))$ *output bits.*

[1] To the matter of fact, Rao entitled his construction a "two-source almost extractor" – a suitable name given its small entropy gap.

Note that the entropy gap g is independent of the entropy k and scales *sub*-logarithmically with $1/\varepsilon$. We prove Theorem 5.1.2, whose formal statement is given in Theorem 5.4.6, in two steps. First, we construct a two-source condenser with the same guarantees as provided by Theorem 5.1.2, though only with $m = k^\alpha$ output bits, where $0 < \alpha < 1$ is some small universal constant (see Theorem 5.4.2). This part of the construction is based on our study of entropy-resilient functions (Section 5.3) and on the adaptation of the Chattopadhyay-Zuckerman construction for entropy-resilient functions. To reduce the huge entropy loss we incur (i.e., to increase the output length from k^α to $k - O(\log(1/\varepsilon))$), in the second step, we construct a seedless condenser for block-sources – a result that we believe is of independent interest on which we now elaborate.

5.1.3 Seedless condensers for a single block-source

A (k_1, k_2) *block-source* is a pair of random variables X_1, X_2 that, although may be dependent, have the following guarantee. First, X_1 is a k_1-source, and second, conditioned on any fixing of X_1, the random variable X_2 has min-entropy k_2. Throughout this section, we denote the length of X_1 by n_1 and the length of X_2 by n_2. Informally, the notion of a block-source "lies between" a single source and two independent sources. Indeed, any (k_1, k_2) block-source is a $(k_1 + k_2)$-source. Moreover, if X_1 is a k_1-source and X_2 is an independent k_2-source then X_1, X_2 is a (k_1, k_2) block-source.

Block-sources are key to almost all constructions of seeded extractors as well as to the construction of Ramsey graphs. As mentioned above, there is no one-source extractor, whereas two-source extractors exist even for very low min-entropy. Despite being more structured than a general source, it is a well-known fact that there is no extractor for a single block-source (with non-trivial parameters).

A key component that allows us to increase the output length of our condenser discussed above is a seedless condenser for a single block-source. Let X_1, X_2 be a (k_1, k_2) block-source. Write $g = n_2 - k_2$ for the entropy gap of X_2. For any given $\varepsilon > 0$, we show how to *deterministically* transform X_1, X_2 to a single m-bit random variable, where $m = k_1 - g - O(\log(1/\varepsilon))$, that is ε-close to having min-entropy $m - g - 1$. That is, informally, we are able to condense X_1 roughly to its entropy content k_1 using (the dependent random variable) X_2 while inheriting the entropy gap of X_2 both in the resulted entropy gap and entropy loss. We stress that this transformation is deterministic. This demonstrates that despite the well-known fact that a block-source extractor does not exist, a block-source condenser does. For a formal treatment, see Section 5.4.3.

5.1.4 A three-source extractor

An immediate implication of Theorem 5.1.2 are low-error three-source extractors supporting min-entropies $k_1 = k_2 = \mathsf{polylog}(n/\varepsilon)$ and $k_3 = \Omega(\log(1/\varepsilon))$. This is achieved by feeding our condenser's output $Y = \mathsf{Cond}(X_1, X_2)$ as a seed to a seeded extractor that supports small entropies (see, e.g., Theorem 2.2.3), outputting $\mathsf{Ext}(X_3, Y)$. We can compensate for the tiny entropy gap of Y by employing Ext with a slightly lower error.

Corollary 5.1.3. *For all integers n, k, k' and every $\varepsilon > 0$ such that $n \geq k \geq \mathsf{polylog}(\frac{n}{\varepsilon})$ and $n \geq k' \geq \Omega(\log \frac{1}{\varepsilon})$ there exists a $\mathsf{poly}(n, \log(1/\varepsilon))$-time computable three-source extractor*

$$\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$$

for min-entropies k, k, k' and error guarantee ε, where $m = k' - O(\log(1/\varepsilon))$.

We note that when ε is taken sub-polynomial in n, in which case the two-source extractor of [CZ16] is not polynomial-time computable, Corollary 5.1.3 modestly improves upon known three-source extractors that either require all three-sources to have min-entropy $\mathsf{polylog}(\frac{n}{\varepsilon})$ [Li15b] or require, for any parameter $\delta > 0$, min-entropies δn, $\mathsf{poly}(\frac{1}{\delta}) \log(\frac{n}{\varepsilon})$, $\mathsf{poly}(\frac{1}{\delta}) \log(\frac{\log n}{\varepsilon})$ [Coh16a].

5.2 Preliminaries

5.2.1 Two-source condensers

Definition 5.2.1. *A function*

$$\mathsf{Cond}\colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$$

is an $((n_1, k_1), (n_2, k_2)) \to_\varepsilon (m, k' = m - g)$ two-source condenser *if the following holds. For every (n_1, k_1) source X_1 and an independent (n_2, k_2) source X_2, the output $\mathsf{Cond}(X_1, X_2)$ is ε-close to an (m, k') source. We refer to ε as the* error guarantee *and to g as the* entropy gap *of* Cond. ◊

Definition 5.2.2. *A function*

$$\mathsf{Cond}\colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$$

is a strong *$((n_1, k_1), (n_2, k_2)) \to_{\varepsilon_1, \varepsilon_2} (m, k')$* two-source condenser (in the first source) *if the following holds. For every (n_1, k_1) source X_1 and an independent (n_2, k_2) source X_2, with probability $1 - \varepsilon_1$ over $x_1 \sim X_1$, the output $\mathsf{Cond}(x_1, X_2)$ is ε_2-close to an (m, k') source.* ◊

Similarly, one can define, in the natural way, a condenser that is strong in the second source.

5.2.2 Fooling AC circuits

A Boolean circuit is an $\mathsf{AC}[d, s]$ circuit if it has depth d, size s and unbounded fan-in. We say that a circuit C with n input bits is *ε-fooled* by a distribution D if $|C(D) - D(U_n)| \leq \varepsilon$.

Harsha and Srinivasan [HS16], improving upon Braverman's seminal result [Bra10] (see also [Tal17]) proved:

Theorem 5.2.3 ([HS16])**.** *There exists a constant $c > 0$ such that the following holds. For all integers s, d, t, any $\mathsf{AC}[d, s]$ circuit is ε-fooled by any t-wise independent distribution, where $\varepsilon = 2^{(\log s)^{3d+c} - t}$.*

Definition 5.2.4. *Let X be a distribution over $\{0,1\}^n$. We say* $\mathrm{bias}(X) \leq \varepsilon$ *if for every non-empty $S \subseteq [n]$,* $\Pr\left[\bigoplus_{i \in S} X_i = 1\right] \in \left[\frac{1}{2} \pm \varepsilon\right]$. ◊

Lemma 5.2.5 ([Vaz86])**.** *Let X be a distribution over $\{0,1\}^n$. Then, $|X - U_n| \leq 2^{n/2} \cdot \mathrm{bias}(X)$.*

We need a slight generalization of Theorem 5.2.3.

Lemma 5.2.6. *There exists a constant $c > 0$ such that the following holds for all integers n, m, d, s. Let $C\colon \{0,1\}^n \to \{0,1\}^m$ be an $\mathsf{AC}[d,s]$ circuit. Then, C is ε-fooled by any t-wise independent distribution, where $\varepsilon = 2^{(m+\log s)^{3d+c}-t}$.*

Proof. Fix $S \subseteq [m]$ and consider the circuit $C_S\colon \{0,1\}^n \to \{0,1\}$ given by

$$C_S(x) = \bigoplus_{i \in S} C(x)_i.$$

As the parity over $|S|$ bits can be expressed as a CNF (or DNF) of at most $2^{|S|}$ terms, C_S is an $\mathsf{AC}[d+1, s']$ circuit, for $s' = O(s + m2^m)$. By Theorem 5.2.3, every t-wise distribution ε'-fools C_S, where $\varepsilon' = 2^{(m+\log s)^{3d+c'}-t}$ for some universal constant $c' > 0$. That is, for every t-wise distribution D and non-empty $S \subseteq [m]$, $|C_S(D) - C_S(U_n)| \leq \varepsilon'$. By Lemma 5.2.5,

$$\varepsilon = |C(D) - C(U_n)| \leq 2^{m/2}\varepsilon' = 2^{(m+\log s)^{3d+c}-t}$$

for some universal constant $c > c' > 0$. □

5.3 Entropy-Resilient Functions

We extend the notion of non-oblivious *bit*-fixing sources from Section 2.1.3 to Σ-fixing sources.

Definition 5.3.1. *Let $\Sigma = \{0,1\}^m$. A (q,t)* non-oblivious Σ-fixing source *$X = (X_1, \ldots, X_r)$ is a random variable over $\Sigma^r = \{0,1\}^{rm}$ for which there exists a set $R_{\mathsf{bad}} \subseteq [r]$ of cardinality $q' \leq q$ such that:*

- *The joint distribution of $\{(X_i)_j \mid i \in [r] \setminus R_{\mathsf{bad}}, j \in [m]\}$, denoted by G_X, is t-wise independent over $\{0,1\}^{(r-q')m}$; and*
- *Each of the random variables in $B_X \triangleq \{(X_i)_j\}$ with $i \in R_{\mathsf{bad}}$ and $j \in [m]$ may depend arbitrarily on all other random variables in G_X and B_X.*

If $t = (r-q')m$ we say X is a q-non-oblivious Σ-fixing source. If $m = 1$ we say X is a bit-fixing source and the definition coincides with the standard definition of non-oblivious bit-fixing sources. When X is clear from context, we write G and B for G_X and B_X, respectively. ◊

Definition 5.3.2. *Let $\Sigma = \{0,1\}^m$. A function $f\colon \Sigma^r \to \Sigma$ is a (q,t,g,ε)* entropy-resilient function *if for every (q,t) non-oblivious Σ-fixing source X over Σ^r, the output $f(X)$ is ε-close to an $(m, m-g)$ source. If $g = 0$ we say f is (q,t,ε) resilient.* ◊

5.3.1 Functions with one output bit

Definition 5.3.3. *Let $f\colon \{0,1\}^r \to \{0,1\}$ be an arbitrary function. Let X be a (q,t) non-oblivious bit-fixing source over $\{0,1\}^r$. Define $E(f)$ to be the event in which the bits tossed by the good players do not determine the value of the function f. We define the influence of the bad players by $I(f) = \Pr[E(f)]$.* ◊

Previously, we mentioned that Chattopadhyay and Zuckerman [CZ16], followed by an improvement by Meka [Mek17], derandomized the Ajtai-Linial function [AL93]. We rephrase Theorem 2.1.15 using our new notations, and we will also need the fact that their almost-balanced resilient function which is also computable by monotone AC^0 circuits.

Theorem 5.3.4 ([CZ16, Mek17])**.** *For every constant $0 < \delta < 1$, there exists a constant $c_\delta \geq 1$ such that for every constant $c \geq c_\delta$ and integer r there exists a monotone function $\mathsf{Res}\colon \{0,1\}^r \to \{0,1\}$ such that for every $t \geq c\log^4 r$,*

- *For every (q,t)-non-oblivious bit-fixing source X, $I(\mathsf{Res}) \leq c \cdot \frac{q}{r^{1-\delta}}$.*
- *For every t-wise independent distribution D, $\mathrm{bias}(\mathsf{Res}(D)) \leq r^{-1/c}$.*

The function Res is computable by a uniform depth 3 monotone circuit of size r^c. Further, the function $c_\delta(\delta)$ is continuous and monotonically decreasing.

We also make use of the following corollary.

Corollary 5.3.5. *For every constant $0 < \gamma < 1$ there exist constants $0 < \alpha < \beta < 1$ such that for every integer r there exists a function $\mathsf{Res}\colon \{0,1\}^r \to \{0,1\}$ which for every $t \geq \frac{1}{\beta}\log^4 r$ satisfies: For every $(r^{1-\gamma}, t)$ non-oblivious bit-fixing source X,*

$$\begin{aligned} I(\mathsf{Res}) &\leq \frac{1}{\beta}\cdot r^{-\alpha}, \\ \mathrm{bias}\left(\mathsf{Res}(X) \mid \neg E(\mathsf{Res})\right) &\leq \frac{3}{\beta}\cdot r^{-\alpha}. \end{aligned}$$

The function Res is computable by a uniform depth 3 monotone circuit of size $r^{\frac{1}{\beta}}$.

Proof. Using the notations of Theorem 5.3.4, assume that for every η, $c_\eta > \frac{1}{2\eta}$ (if not, we can always increases c_η). Given $\gamma > 0$, set δ to be the constant satisfying the equation $f(\delta) = \delta - \gamma + \frac{1}{2c_\delta} = 0$. Such a δ exists, as $f(\delta) \leq 2\delta - \gamma$ and therefore $f(\delta) < 0$ when δ approaches 0, and $f(\delta) > 0$ when δ approaches γ. Note that by our choice of δ, it holds that

$$\delta < \gamma = \delta + \frac{1}{2c_\delta} < \delta + \frac{1}{c_\delta}.$$

Set $\alpha = \gamma - \delta > 0$ and $\beta = \frac{1}{c_\delta}$. Note that indeed $\beta > \alpha$.

By Theorem 5.3.4, applied with the constant δ, it holds that $I(\mathsf{Res}) \leq c_\delta \frac{r^{1-\gamma}}{r^{1-\delta}} = \frac{1}{\beta} r^{-\alpha}$. Further, $\mathrm{bias}(\mathsf{Res}(D)) \leq r^{-\beta}$.

Following similar arguments as in [CZ16], we have that $\mathrm{bias}(\mathsf{Res}(X)) \leq \frac{1}{\beta} r^{-\alpha} + r^{-\beta}$, so

$$\mathrm{bias}\left(\mathsf{Res}(X) \mid \neg E(\mathsf{Res})\right) \leq \frac{\frac{1}{\beta} r^{-\alpha} + r^{-\beta}}{1 - \frac{1}{\beta} r^{-\alpha}} \leq \frac{3}{\beta} r^{-\alpha}.$$

□

5.3.2 Functions with multiple output bits

The output bit of a (q, t, ε) resilient function $f\colon \{0,1\}^r \to \{0,1\}$ applied to a (q, t) non-oblivious bit-fixing source is indeed ε-close to uniform, but, as shown by [KKL88] even when $q = 1$, ε cannot be smaller that $\frac{\ln r}{r}$ (and the simpler bound $\varepsilon \geq \frac{1}{r}$ is almost trivial). We show that when we output many bits, and allow $o(\log \frac{1}{\varepsilon})$ entropy gap, we may obtain much smaller error. We do that by exhibiting an entropy-resilient function based on a parallel application of the (derandomized version of the) Ajtai-Linial function.

A construction of an entropy-resilient function. Given a constant $0 < \gamma < 1$ and integers $r \geq m$ let $\mathsf{Res}\colon \{0,1\}^r \to \{0,1\}$ be the function guaranteed by Corollary 5.3.5 with respect to γ. Define $\Sigma = \{0,1\}^m$ and

$$\mathsf{EntRes}\colon \Sigma^r \to \Sigma$$

as follows. On input $x \in \Sigma^r$,

$$\mathsf{EntRes}(x) = (\mathsf{Res}(x_1), \ldots, \mathsf{Res}(x_m)),$$

where x_i stands for the i-th column of x, when we view x as a $r \times m$ table.

Theorem 5.3.6. *For every constant $0 < \gamma < 1$ there exist constants $0 < \alpha < \beta < 1$ and $c' \geq 1$ such that the following holds. For all integers r, $m \leq r^{\alpha/2}$, every $\varepsilon > 0$, and every integer $t \geq (m + \log(r/\varepsilon))^{c'}$, the function $\mathsf{EntRes}\colon \Sigma^r \to \Sigma$ is $(q = r^{1-\gamma}, t, g, \varepsilon)$ entropy-resilient with entropy gap $g = o(\log(1/\varepsilon))$.*

The proof of Theorem 5.3.6 is done in two steps. First, in Section 5.3.2.1, we analyze the theorem for the special case in which the distribution G_X of the given non-oblivious Σ-fixing source X is uniform. Then, based on that result, in Section 5.3.2.2 we prove Theorem 5.3.6.

5.3.2.1 The uniform case

In this section, we prove the following lemma.

Lemma 5.3.7. *Keeping the notations of Theorem 5.3.6, the function $\mathsf{EntRes}\colon \Sigma^r \to \Sigma$ is $(q = r^{1-\gamma}, g, \varepsilon)$ entropy-resilient with entropy gap*

$$g = c_{\mathsf{ent}_1} \frac{\ln \frac{1}{\varepsilon}}{\ln \ln \frac{1}{\varepsilon} + c_{\mathsf{ent}_2} \ln r} = o(\log(1/\varepsilon))$$

for some universal constant $c_{\mathsf{ent}_1} > 0$ and a constant $c_{\mathsf{ent}_2} > 0$ that depends only on γ.

Proof. Let X be a $q = r^{1-\gamma}$ non-oblivious Σ-fixing source. Let $R_{\mathsf{bad}} \subseteq [r]$ be the set of bad players, and E_i the event that the values of the good players in X_i do not determine the value of Res. Note that we shall also denote E_i as an indicator for that event. By Corollary 5.3.5, there exists constants $0 < \alpha < \beta < 1$ such that

$$\Pr[E_i = 1] \leq \frac{1}{\beta} \cdot r^{-\alpha}$$

for every $i \in [m]$. Observe that the random variables $E_1, \ldots, E_m$ are independent, as the value of E_i depends only on the values of the good players in the i-th column, and by assumption all these values are independent of the corresponding values in the other columns. Write

$$\mu = m \cdot \frac{1}{\beta} \cdot r^{-\alpha}$$

and note that since $m \leq r^{\alpha/2}$, $\mu < 1$. Set

$$c = \frac{4 \ln \frac{1}{\varepsilon}}{\mu} \cdot \frac{1}{\ln \frac{\ln \frac{1}{\varepsilon}}{\mu}}$$

and observe that $c > 1$. By the Chernoff bound,

$$\Pr\left[\sum_{i=1}^{m} E_i > c\mu\right] \leq \left(\frac{e^{c-1}}{c^c}\right)^{\mu} \leq e^{-\frac{1}{2}\mu c \ln c} \leq \varepsilon,$$

where the last inequality follows from the fact that $c \ln c \geq \frac{2 \ln \frac{1}{\varepsilon}}{\mu}$.

By Corollary 5.3.5, for every $i \in [m]$,

$$\text{bias}\left(\mathsf{Res}(X_i) \mid E_i = 0\right) \leq \frac{3}{\beta} \cdot r^{-\alpha}.$$

Assume that the event $\sum_{i=1}^{m} E_i \leq c\mu$ holds, and let $I \subseteq [m]$, $|I| \geq m - c\mu$ be the set of good columns I for which $E_i = 0$. For every $w \in \{0,1\}^m$, we have:

$$\begin{aligned}
\Pr[\mathsf{EntRes}(X) = w] &\leq \Pr[\mathsf{EntRes}(X)_I = w_I] \\
&\leq \left(\frac{1}{2} + \frac{3}{\beta} \cdot r^{-\alpha}\right)^{m - c\mu} \\
&\leq 2^{-m + c\mu} e^{\frac{6}{\beta} r^{-\alpha} m} \\
&\leq 2^{-m + c\mu} 2^{10\mu}.
\end{aligned}$$

Now, we have

$$c\mu + 10\mu \;\leq\; 2c\mu \;\leq\; \frac{8 \ln \frac{1}{\varepsilon}}{\ln \ln \frac{1}{\varepsilon} + \ln \frac{1}{\mu}} \;\leq\; \frac{8 \ln \frac{1}{\varepsilon}}{\ln \ln \frac{1}{\varepsilon} + \frac{4}{\alpha} \ln r} \;=\; o\left(\log \frac{1}{\varepsilon}\right).$$

We have shown that except with probability ε, the output $\mathsf{EntRes}(X)$ has min-entropy $m - o(\log(1/\varepsilon))$, as desired. More specifically, the min-entropy in the good columns alone is at least $m - o(\log(1/\varepsilon))$, and we stress that the good columns are not fixed but depend on the sample itself. □

5.3.2.2 The bounded-independence case – proof of Theorem 5.3.6

Throughout this section, we use the same notations as in Lemma 5.3.7. We are given X that is a (q,t) non-oblivious Σ-fixing source. We use a similar approach to the one taken in [CZ16]. For the sake of the proof, we:

- Let GU be the distribution in which the good players are jointly uniform, and the bad players are arbitrary.
- Define a small-depth circuit C' that is related to EntRes so that $H_\infty(\mathsf{EntRes}(X)) \geq H_\infty(C'(X))$.

We will show that $C'(X)$ and $C'(GU)$ are statistically close to each other. Finally, the results of Section 5.3.2.1 proves that except for a small probability, $H_\infty(C'(GU)) \geq m - o(\log(1/\varepsilon))$.

Proof of Theorem 5.3.6. Fix a (q,t) non-oblivious Σ-fixing source X. Let GU be the distribution where the good players are jointly uniform, and the bad players are arbitrary. We construct a circuit $C' \colon \{0,1\}^{rm} \to \{0,1\}^m$ such that:

$$(C'(x))_i \quad = \quad \begin{cases} \mathsf{EntRes}(x)_i & \text{If } E_i(x) = 0, \\ 0 & \text{Otherwise.} \end{cases}$$

Recall that E_i is *fully determined* by the good players, and so does $\mathsf{EntRes}(X)_i$ when $E_i = 0$. Hence, C' is fully determined by the good players.

We can write a small-depth circuit computing C'. Let C be the depth-3 size $r^{1/\beta}$ circuit that computes the function $\mathsf{Res} \colon \{0,1\}^r \to \{0,1\}$ as guaranteed by Theorem 5.3.4. Construct a circuit for C' as follows:

- For $i \in [m]$ and $b \in \{0,1\}$ let $C_{i,b}$ be a copy of C where we wire $(x_i)_j$ for every good player $j \in [r]$, and the value b for every bad player.
- The top part contains m comparators, outputting the output of $C_{i,0}$ if the output of $C_{i,0}$ is the same as the output of $C_{i,1}$, and 0 otherwise.

The circuit has depth 4 and size $s'' = O(mr^{1/\beta})$ and its correctness is guaranteed by the fact that Res is monotone (so it is sufficient to consider the case where the bad players voted unanimously).

By Lemma 5.2.6, $|C'(GU) - C'(X)| \leq 2^{(m+\log r)^{c''}-t}$ for some large enough universal constant $c'' > 0$. For every $w \in \{0,1\}^m$:

$$\begin{aligned} \Pr\left[\mathsf{EntRes}(X) = w\right] &\leq \Pr\left[\mathsf{EntRes}(X)_I = w_I\right] \\ &= \Pr\left[C'(X)_I = w_I\right] \\ &\leq \Pr\left[C'(GU)_I = w_I\right] + 2^{(m+\log r)^{c''}-t} \\ &\leq 2^{-m+\frac{8\ln\frac{1}{\varepsilon}}{\ln\ln\frac{1}{\varepsilon}+\frac{4}{\alpha}\ln r}} + 2^{(m+\log r)^{c''}-t}, \end{aligned}$$

where in the last inequality we have used Lemma 5.3.7. We can set the constant c' stated in the theorem to be larger than c'' and get that

$$\Pr\left[\mathsf{EntRes}(X) = w\right] \leq 2 \cdot 2^{-m+\frac{8\ln\frac{1}{\varepsilon}}{\ln\ln\frac{1}{\varepsilon}+\frac{4}{\alpha}\ln r}}.$$

To conclude, note that the above holds with probability at least $1-\varepsilon$, and then $\mathsf{EntRes}(X)$ has min-entropy at least $-1 + m - \frac{8\ln\frac{1}{\varepsilon}}{\ln\ln\frac{1}{\varepsilon}+\frac{4}{\alpha}\ln r} = m - o(\log(1/\varepsilon))$, as desired. □

5.4 Low-Error Two-Source Condensers

In Chapter 3 we saw that the [CZ16] framework uses a reduction from two independent sources to non-oblivious bit-fixing sources. In Section 5.4.1 we extend this to many output bits and show a reduction from two independent sources to non-oblivious Σ-fixing sources. In Section 5.4.2 we use this together with the results of Section 5.3 to get a low-error two-source condenser with many output bits, yet still far from getting almost all of the possible entropy from the two sources. In Section 5.4.4 we show how the condenser obtained in Section 5.4.2 can be used to extract more bits and get a condenser extracting almost all the entropy from one of the sources. To this end, we use the connection between condensers with small entropy gap and samplers with multiplicative error (Section 5.4.3) and ideas from [RRV99].

5.4.1 From two independent sources to a non-oblivious Σ-fixing source

In this section, we revisit the [CZ16] transformation of two independent sources to a non-oblivious bit-fixing source, and extend it to sources with several bits. Throughout this section, we refer to $c_{\mathsf{GUV}}, c_{\mathsf{nm}}$ as the constants that appear in Theorem 2.2.3 and Theorem 2.4.5, respectively.

Theorem 5.4.1. *For all integers n, t, m, k, with $n \geq k \geq (tm \log n)^5$ and set $\Sigma = \{0,1\}^m$, there exists a $\mathsf{poly}(n)$-time computable function*

$$\mathsf{TwoSourcesToNOF}\colon \{0,1\}^n \times \{0,1\}^n \to \Sigma^r,$$

where $r = n^{2c_{\mathsf{GUV}}}$ such that the following holds. Let X_1, X_2 be a pair of independent (n,k) sources. Then, with probability at least $1 - 2^{-k/2}$ over $x_2 \sim X_2$, the output

$$\mathsf{TwoSourcesToNOF}(X_1, x_2)$$

is (n^{-mt})-close to an $\left(r^{1-\frac{1}{4c_{\mathsf{GUV}}}}, t\right)$ non-oblivious Σ-fixing source.

Proof. We start by setting the following parameters:

Setting of parameters.

- Set $\varepsilon_{\mathsf{GUV}} = \frac{1}{n}$.
- Set $d_{\mathsf{GUV}} = c_{\mathsf{GUV}} \log\left(\frac{n}{\varepsilon_{\mathsf{GUV}}}\right) = 2c_{\mathsf{GUV}} \log n$.
- Set $\varepsilon_{\mathsf{nm}} = 2^{-4mt(d_{\mathsf{GUV}} + \log m)}$.
- Set $d_{\mathsf{nm}} = c_{\mathsf{nm}} t^2 \log^2\left(\frac{n}{\varepsilon_{\mathsf{nm}}}\right)$.

Note that $\varepsilon_{\mathsf{nm}} = 2^{-\Theta(mt \log n)}$ and that $d_{\mathsf{nm}} = \Theta(t^4 m^2 \log^2 n)$.

Building blocks. For the construction of $\mathsf{TwoSourcesToNOF}$, we make use of the following ingredients:

- Let $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^{d_{\mathsf{GUV}}} \to \{0,1\}^{d_{\mathsf{nm}}}$ be the strong $(k/2, \varepsilon_{\mathsf{GUV}})$ seeded extractor, guaranteed by Theorem 2.2.3. One can verify that $k/2 \geq 2d_{\mathsf{nm}}$ as required by Theorem 2.2.3.
- Let $\mathsf{nmExt}\colon \{0,1\}^n \times \{0,1\}^{d_{\mathsf{nm}}} \to \{0,1\}^m$ be the $(k, \varepsilon_{\mathsf{nm}})$ t-non-malleable extractor, guaranteed by Theorem 2.4.5. Note that $k \geq 3tm$ so the hypothesis of Theorem 2.4.5 is easily met with our choice of parameters.

The construction. We identify $[r]$ with $\{0,1\}^{d_{\mathsf{GUV}}}$. On inputs $x_1, x_2 \in \{0,1\}^n$, we define $\mathsf{TwoSourcesToNOF}(x_1, x_2)$ to be the $r \times m$ matrix whose i-th row is given by

$$\mathsf{TwoSourcesToNOF}(x_1, x_2)_i = \mathsf{nmExt}(x_1, \mathsf{Ext}(x_2, i)).$$

Analysis. Write $D_{\mathsf{nm}} = 2^{d_{\mathsf{nm}}}$ and identify $[D_{\mathsf{nm}}]$ with $\{0,1\}^{d_{\mathsf{nm}}}$. Let $G \subseteq [D_{\mathsf{nm}}]$, $|G| \geq (1 - \sqrt{\varepsilon_{\mathsf{nm}}})D_{\mathsf{nm}}$, be the set of good seeds guaranteed by Lemma 2.4.2. By Lemma 2.1.2, for any distinct $r_1, \ldots, r_t \in G$,

$$(\mathsf{nmExt}(X_1, r_1), \ldots, \mathsf{nmExt}(X_1, r_t)) \approx_{t\sqrt{\varepsilon_{\mathsf{nm}}}} U_{tm}.$$

Let $S(X_2) = \{\mathsf{Ext}(X_2, 1), \ldots, \mathsf{Ext}(X_2, D_{\mathsf{nm}})\}$. By Theorem 2.2.9,

$$\Pr_{x_2 \sim X_2}[|S(x_2) \cap G| \leq (1 - \sqrt{\varepsilon_{\mathsf{nm}}} - \varepsilon_{\mathsf{GUV}}) \cdot r] \leq 2^{-k/2}.$$

We say that $x_2 \in \operatorname{Supp} X_2$ is good if it induces a good sample, that is if $|S(x_2) \cap G| > (1 - \sqrt{\varepsilon_{\mathsf{nm}}} - \varepsilon_{\mathsf{GUV}})r$. Fix a good x_2 and let $Z = \mathsf{TwoSourcesToNOF}(X_1, x_2)$. In the good seeds, every t elements of Z are $(t\sqrt{\varepsilon_{\mathsf{nm}}})$-close to uniform, and there are at most $q \leq (\sqrt{\varepsilon_{\mathsf{nm}}} + \varepsilon_{\mathsf{GUV}})r$ bad rows. Applying Lemma 2.1.11, we get that Z is $\zeta = t\sqrt{\varepsilon_{\mathsf{nm}}}(rm)^{mt}$ to a (q, t) non-oblivious bit-fixing source. By our choice of $\varepsilon_{\mathsf{nm}}$,

$$\zeta = 2^{-2mt(d_{\mathsf{GUV}} + \log m)} 2^{mt \log(rm)} \leq 2^{-mt \log r} \leq n^{-mt}.$$

Further,

$$q \leq (\sqrt{\varepsilon_{\mathsf{nm}}} + \varepsilon_{\mathsf{GUV}})r \leq 2\varepsilon_{\mathsf{GUV}} r = 2r^{-\frac{1}{2c_{\mathsf{GUV}}}+1} \leq r^{1-\frac{1}{4c_{\mathsf{GUV}}}}.$$

We now analyse the running-time. We first apply Ext to compute $S(x_2)$, which takes time $\mathsf{poly}(n, \log(1/\varepsilon_{\mathsf{GUV}})) = \mathsf{poly}(n)$. Then, applying each nmExt takes

$$\mathsf{poly}(n, \log(1/\varepsilon_{\mathsf{nm}})) = \mathsf{poly}(n, m, t, d_{\mathsf{GUV}}) = \mathsf{poly}(n)$$

time and we do it for $r = \mathsf{poly}(n)$ times. Overall, the running time is $\mathsf{poly}(n)$, as required. In particular, as $n \geq k \geq m$, the running time is also poly-logarithmic in the errors of the construction, $2^{-k/2}$ and n^{-mt}. □

5.4.2 Low-error condensers with high entropy loss

Theorem 5.4.2. *There exists a constant $c \geq 1$ such that the following holds. For all integers n, k, m and every $\varepsilon > 0$ such that $n \geq k \geq (m \log(n/\varepsilon))^c$ there exists a $\mathsf{poly}(n)$-time computable $((n,k),(n,k)) \to_{\varepsilon, 2^{-k/2}} (m, m-g)$ two-source condenser*

$$\mathsf{Cond}' \colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m ,$$

that is strong in the second source, with entropy gap $g = o(\log(1/\varepsilon))$.

Proof. We start by describing the construction of our condenser Cond' and then turn to the analysis. As usual, we let c_{GUV} be the constant that is given by Theorem 2.2.3.

Setting of parameters.

- Set $\gamma = \frac{1}{4c_{\mathsf{GUV}}}$ and let $0 < \alpha < \beta < 1$ and c' be the constants from Theorem 5.3.6 with respect to this γ.
- Set $r = n^{2c_{\mathsf{GUV}}}$.
- Set $t = (m + \log(r/\varepsilon))^{c'}$.
- Set c, the constant stated in this theorem, to $c = \max(10c', 2/\alpha)$.

Building blocks.

- Let $\mathsf{TwoSourcesToNOF} \colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{r \times m}$ be the function that is given by Theorem 5.4.1. We are about to apply $\mathsf{TwoSourcesToNOF}$ to (n,k) sources, and indeed k is large enough to satisfy the hypothesis of Theorem 5.4.1.
- Let $\mathsf{EntRes} \colon \{0,1\}^{r \times m} \to \{0,1\}^m$ be the function from Theorem 5.3.6 when set with the parameter γ as defined above. Note that the hypothesis of Theorem 5.3.6 holds, as since $c \geq 2/\alpha$ we have that $m < r^{\alpha/2}$, and t is large enough.

The construction. On inputs $x_1, x_2 \in \{0,1\}^n$, we define

$$\mathsf{Cond}'(x_1, x_2) = \mathsf{EntRes}\left(\mathsf{TwoSourcesToNOF}(x_1, x_2)\right).$$

Analysis. Clearly, EntRes is computable in $\mathsf{poly}(m, r) = \mathsf{poly}(n)$ time. Let X_1, X_2 be a pair of independent (n,k)-sources. By Theorem 5.4.1, except with probability $2^{-k/2}$ over $x_2 \sim X_2$, the output $\mathsf{TwoSourcesToNOF}(X_1, x_2)$ is n^{-mt}-close to an $(r^{1-\gamma}, t)$ non-oblivious bit-fixing source. For every x_2 for which this event holds, the output

$$\mathsf{EntRes}(\mathsf{TwoSourcesToNOF}(X_1, x_2))$$

is $(n^{-mt} + \varepsilon)$-close to an $(m, m - o(\log(1/\varepsilon)))$ source, and $n^{-mt} \leq \varepsilon$. □

5.4.3 Deterministically condensing a single block-source

A distribution (X, Y) is a blockwise source if both X has sufficient min-entropy and also for every $x \in \operatorname{Supp} X$, $(Y \mid X = x)$ has sufficient min-entropy.

Lemma 5.4.3. *Let X be an (n, k) source. Let Y be a d-bit random variable (that may depend on X) such that for every $x \in \operatorname{Supp} X$, the random variable $(Y \mid X = x)$ is ε_{B}-close to a $(d, d - g)$ source.*

Let $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a $(k_{\mathsf{Ext}}, \varepsilon_{\mathsf{Ext}})$ seeded extractor. Suppose $k \geq k_{\mathsf{Ext}} + \log(1/\varepsilon_{\mathsf{Ext}})$. Then, $\mathsf{Ext}(X, Y)$ is $(2^{g+2}\varepsilon_{\mathsf{Ext}} + 2\varepsilon_{\mathsf{B}})$-close to an $(m, m - g - 1)$ source.

Proof. Fix any $T \subseteq \{0,1\}^m$. Define the set

$$\mathsf{OverHit}_T = \left\{ x \in \{0,1\}^n : \Pr_{y \sim U_d}[\mathsf{Ext}(x, y) \in T] > \rho(T) + \varepsilon_{\mathsf{Ext}} \right\}.$$

Claim 5.4.4. $|\mathsf{OverHit}_T| < 2^{k_{\mathsf{Ext}}}$.

Proof. Suppose towards a contradiction that $|\mathsf{OverHit}_T| \geq 2^{k_{\mathsf{Ext}}}$ and let B denote the random variable that is uniform over the set $\mathsf{OverHit}_T$. Since B has min-entropy at least k_{Ext}, the output $\mathsf{Ext}(B, U_d)$ is $\varepsilon_{\mathsf{Ext}}$-close to uniform, and therefore $\Pr_{x \sim B, y \sim U_d}[\mathsf{Ext}(x, y) \in T] \leq \rho(T) + \varepsilon_{\mathsf{Ext}}$. This stands in contradiction to the definition of B. □

Now,

$$\Pr[\mathsf{Ext}(X, Y) \in T] \leq \Pr[\mathsf{Ext}(X, Y) \in T \mid X \notin \mathsf{OverHit}_T] + \Pr[X \in \mathsf{OverHit}_T].$$

By Claim 5.4.4, $\Pr[X \in \mathsf{OverHit}_T] \leq 2^{k_{\mathsf{Ext}} - k}$. Also, for every $x \notin \mathsf{OverHit}_T$ let

$$GY_x = \left\{ y \in \{0,1\}^d : \mathsf{Ext}(x, y) \in T \right\}.$$

By definition, $\rho(GY_x) \leq \rho(T) + \varepsilon_{\mathsf{Ext}}$. Therefore,

$$\Pr_{y \sim (Y \mid X = x)}[\mathsf{Ext}(x, y) \in T] = \Pr_{y \sim (Y \mid X = x)}[y \in GY_x] \leq \varepsilon_{\mathsf{B}} + |GY_x| 2^{g-d} \leq \varepsilon_{\mathsf{B}} + 2^g(\rho(T) + \varepsilon_{\mathsf{Ext}}).$$

Thus,

$$\begin{aligned}
\Pr[\mathsf{Ext}(X, Y) \in T] &\leq \Pr[\mathsf{Ext}(X, Y) \in T \mid X \notin \mathsf{OverHit}_T] + \Pr[X \in \mathsf{OverHit}_T] \\
&\leq 2^g \rho(T) + 2^g \varepsilon_{\mathsf{Ext}} + \varepsilon_{\mathsf{B}} + 2^{k_{\mathsf{Ext}} - k} \\
&\leq 2^g \rho(T) + (2^g + 1)\varepsilon_{\mathsf{Ext}} + \varepsilon_{\mathsf{B}}.
\end{aligned}$$

But,

Claim 5.4.5. *Let Z be a random variable over n-bit strings such that for every $T \subseteq \{0,1\}^n$, $\Pr[Z \in T] \leq 2^g \rho(T) + \varepsilon$. Then, Z is 2ε-close to an $(n, n - g - 1)$ source.*

Proof. Set $H = \left\{x : \Pr[X = x] > 2^{-(n-g-1)}\right\}$. On the one hand,

$$\Pr[Z \in H] = \sum_{x \in H} \Pr[X = x] \geq 2^{g+1}\rho(H).$$

On the other hand, by our assumption, $\Pr[X \in H] \leq 2^g\rho(H) + \varepsilon$. Together, we get that $2^g\rho(H) \leq \varepsilon$. Thus, $\Pr[X \in H] \leq 2\varepsilon$. As H are all the heavy elements, we conclude that Z is 2ε-close to a distribution with $n - g - 1$ min-entropy. □

We can therefore summarize that $\mathsf{Ext}(X, Y)$ is $(2^{g+2}\varepsilon_{\mathsf{Ext}} + 2\varepsilon_{\mathsf{B}})$-close to an $(m, m - g - 1)$ source. □

5.4.4 Low-error condensers with small entropy gap outputting many bits

In this section we prove our main theorem, that readily implies Theorem 5.1.2.

Theorem 5.4.6. *There exists a constant $c \geq 1$ such that the following holds. For all integers n, k and every $\varepsilon > 0$ such that $k \geq \log^c(n/\varepsilon)$ there exists a $\mathsf{poly}(n, \log(1/\varepsilon))$-time computable $((n, k), (n, k)) \to_\varepsilon (m, m - g)$ two-source condenser*

$$\mathsf{Cond}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$$

with $m = k - 5\log(1/\varepsilon) - O(1)$ and $g = o(\log(1/\varepsilon))$.

Proof. We start by setting some parameters.

Parameters.

- Set $\varepsilon_{\mathsf{Cond}'} = \varepsilon/8$.
- Set $\varepsilon_{\mathsf{Ext}} = \varepsilon^2/32$.
- Set $k_{\mathsf{Ext}} = k - \log(2/\varepsilon)$.
- Set $d_{\mathsf{Ext}} = c' \log n \cdot \log(n/\varepsilon_{\mathsf{Ext}})$ where c' is the constant that is given by Theorem 2.2.4.

For the construction we make use of the following building blocks.

- Let $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^{d_{\mathsf{Ext}}} \to \{0,1\}^m$ be the strong $(k_{\mathsf{Ext}}, \varepsilon_{\mathsf{Ext}})$ seeded extractor that is given by Theorem 2.2.4. By that theorem, $m = k_{\mathsf{Ext}} - 2\log(1/\varepsilon_{\mathsf{Ext}}) - O(1)$.
- Let $\mathsf{Cond}'\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{d_{\mathsf{Ext}}}$ be the $((n, k), (n, k)) \to_{\varepsilon_{\mathsf{Cond}'}, 2^{-k/2}} (d_{\mathsf{Ext}}, d_{\mathsf{Ext}} - g')$ condenser, strong in the second source, that is given by Theorem 5.4.2, with $g' = o(\log(1/\varepsilon_{\mathsf{Cond}'}))$. Note that our choice of parameters satisfies the hypothesis of Theorem 5.4.2 for a large enough constant c.

The construction. On inputs $x_1, x_2 \in \{0,1\}^n$, we define

$$\mathsf{Cond}(x_1, x_2) \;=\; \mathsf{Ext}(x_2, \mathsf{Cond}'(x_1, x_2)).$$

Analysis. Let X_1, X_2 be a pair of independent (n, k) sources. By Theorem 5.4.2, with probability at least $1 - 2^{-k/2}$ over $x_2 \sim X_2$, the random variable $\mathsf{Cond}'(X_1, x_2)$ is $\varepsilon_{\mathsf{Cond}'}$-close to a $(d, d-g')$ source. Lemma 5.4.3 implies that $\mathsf{Ext}(X_2, \mathsf{Cond}'(X_1, X_2))$ is $2^{-k/2} + (2^{g'+2}\varepsilon_{\mathsf{Ext}} + 2\varepsilon_{\mathsf{Cond}'})$-close to an $(m, m - g' - 1)$ source.

By our choice of parameters, $2^{-k/2} + 2^{g'+1}\varepsilon_{\mathsf{Ext}} + 2\varepsilon_{\mathsf{Cond}'} \leq \varepsilon$. Note that $k - m = \log(2/\varepsilon) + 2\log(1/\varepsilon_{\mathsf{Ext}}) = 5\log(1/\varepsilon) + O(1)$. The running-time of the construction readily follows from the running-times of Cond' and Ext. □

Chapter 6

Almost Optimal Erasure List-Decodable Codes

In this chapter we will see that recent developments in extractor theory, some of which outlined in previous chapters, turned out to be beneficial also for other pseudorandomness constructions – erasure list decodable codes and strong seeded dispersers.

A code $\mathcal{C}$ is $(1-\tau, L)$ erasure list-decodable if for every word w, after erasing any $1-\tau$ fraction of the symbols of w, the remaining τ-fraction of its symbols have at most L possible completions into codewords of $\mathcal{C}$.

Non-explicitly, there exist binary $(1-\tau, L)$ erasure list-decodable codes having rate $O(\tau)$ and tiny list-size $L = O(\log \frac{1}{\tau})$. Achieving either of these parameters explicitly is a natural open problem and was brought up in several works (e.g., [GI02, Gur03, Gur04a]). While partial progress on the problem has been achieved, no explicit construction up to this work achieved rate better than $\Omega(\tau^2)$ or list-size smaller than $\Omega(1/\tau)$ (for τ small enough). Furthermore, Guruswami showed that no *linear* code can have list-size small than $\Omega(1/\tau)$ [Gur03]. In this chapter we construct an explicit binary $(1-\tau, L)$ erasure list-decodable code having rate $\tau^{1+\gamma}$ (for any constant $\gamma > 0$ and τ small enough) and list-size $\mathsf{polylog}(\frac{1}{\tau})$, answering simultaneously both questions, and exhibiting an explicit non-linear code that provably beats the best possible linear one.

The binary erasure list-decoding problem is equivalent to the construction of explicit, low-error, strong dispersers outputting one bit with minimal entropy loss and seed-length. Specifically, such dispersers with error ε have an unavoidable entropy loss of $\log\log(\frac{1}{\varepsilon})$ and seed-length at least $\log(\frac{1}{\varepsilon})$. Similarly to the situation with erasure list-decodable codes, no explicit construction achieved seed-length better than $2\log(\frac{1}{\varepsilon})$ or entropy loss smaller than $2\log(\frac{1}{\varepsilon})$, which are the best possible parameters for extractors. For every constant $\gamma > 0$ and every small ε, we explicitly construct an ε-error one-bit strong disperser with near-optimal seed-length $(1+\gamma)\log(\frac{1}{\varepsilon})$ and near-optimal entropy loss $O(\log\log\frac{1}{\varepsilon})$.

The main ingredient in our construction is a new (and almost-optimal) *unbalanced* two-source extractor. The extractor extracts one bit with constant error from two independent sources, where one source has length n and tiny min-entropy $O(\log\log n)$ and the other source has length $O(\log n)$ and arbitrarily small constant min-entropy rate.

6.1 Introduction

Extractors and dispersers, introduced in Chapter 2, differ in the way they measure the proximity of the output distribution to the uniform distribution: Extractors use the total-variation distance, whereas dispersers use support-size distance (that is, they count the number of elements not in the image of the hash function). Thus, extractors are stronger objects than dispersers. Roughly speaking, extractors are needed to derandomize two-sided error algorithms whereas dispersers suffice for one-sided error derandomization.

A function $C\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a *strong* (k,ε) *extractor* if for any k-source X the output distribution $(U_d, C(X,U_d))$, containing the seed y along with output $C(x,y)$, is ε-close to the uniform distribution over $\{0,1\}^d \times \{0,1\}^m$. In contrast, C is a *strong* (k,ε) *disperser* if for any k-source X, the support of $(U_d, C(X,U_d))$ covers at least $(1-\varepsilon)2^{d+m}$ elements from $\{0,1\}^d \times \{0,1\}^m$.

There are two natural parameters measuring the quality of extractors and dispersers:

1. **Seed length.** Both extractors and dispersers use an auxiliary uniform independent source to extract the entropy from the weak source X. The length d of the auxiliary source is called the *seed-length*. We would like the seed-length to be as small as possible.

2. **Entropy loss.** There are $k+d$ bits of entropy in the system: k bits coming from the k-source X, and d bits from the independent uniform seed. The entropy loss is $k-m$, i.e., the difference between the entropy in the input system (including the seed) and the output system (of length $d+m$).

As noted, strong dispersers are weaker objects than strong extractors. The interest in dispersers stems from the fact that their parameters can outperform those of extractors. In Section 2.2 we saw that every strong extractor requires seed-length $d \geq 2\log(\frac{1}{\varepsilon}) + \log(n-k) - O(1)$ and has an unavoidable entropy loss of $k-m \geq 2\log(\frac{1}{\varepsilon}) - O(1)$. Non-explicitly there exist strong extractors with seed-length $d \leq 2\log(\frac{1}{\varepsilon}) + \log(n-k) + O(1)$ and entropy loss $k-m \leq 2\log(\frac{1}{\varepsilon}) + O(1)$. For strong dispersers, [RTS00] showed that every strong disperser requires seed-length $d \geq \log(\frac{1}{\varepsilon}) + \log(n-k) - O(1)$ and has an unavoidable entropy loss $k-m \geq \log\log(\frac{1}{\varepsilon}) - O(1)$. Again, non-explicitly, there exist strong dispersers with seed-length $d \leq \log(\frac{1}{\varepsilon}) + \log(n-k) + O(1)$ and entropy loss $k-m \leq \log\log(\frac{1}{\varepsilon}) + O(1)$ [RTS00, MRZ14].

For strong dispersers, even the case of outputting just one bit in a way that outperforms extractors constructions has been widely open. Indeed, any construction of a disperser with parameters beating those of the best possible extractor must yield a distribution that covers many strings but is necessarily far from uniform, and it is not clear at all how to construct such an object. Gradwohl et al. [GKRTS05] noticed that such strong dispersers imply good Ramsey graphs, another problem that withstood many attempts for many years, until the recent breakthrough result of Chattopadhyay and Zuckerman [CZ16].

In this chapter we go in the reverse direction of that taken in [GKRTS05]. By using the recent machinery of non-malleable extractors and their connection to two-source extractors, we construct near-optimal *unbalanced* two-source extractors (which imply near-optimal *unbalanced* Ramsey graphs). We use these extractors to obtain explicit strong dispersers that output a single bit, with near-optimal seed-length and near-optimal entropy loss.

Theorem 6.1.1 (see also Theorem 6.5.2)**.** *For every constant $0 < \gamma < 1$ and $\varepsilon = n^{-\Omega(1)}$ there exists an explicit strong (k, ε) disperser $\mathsf{Disp}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$ with $d = (1+\gamma)\log(\frac{1}{\varepsilon})$ and $k = O(\log\log\frac{1}{\varepsilon})$, where the constant in the $O(\cdot)$ notation is independent of n but may depend on γ.*

We remark that the dependence of the seed-length on the error is $(1+\gamma)\log(\frac{1}{\varepsilon}) < 2\log(\frac{1}{\varepsilon})$, and the entropy loss is $O(\log\log\frac{1}{\varepsilon}) < 2\log(\frac{1}{\varepsilon})$ and both these bounds are optimal for dispersers up to small factors and are *unattainable* for extractors.

Most previous dispersers constructions have not obtained parameters better than the extractors lower bounds, and we are only aware of one exception: Meka et al. [MRZ14], extending the techniques in [GKRTS05], gave a strong disperser with optimal entropy loss. However, their construction works only for extremely high min-entropy $k = n - \Theta(1)$ and has suboptimal seed-length.

6.1.1 Erasure list-decodable codes

We now turn our attention to binary list-decodable codes in the erasures model. A code $\mathcal{C}$ is a set $\mathcal{C} \subseteq \mathbb{F}_2^n$. We call elements in $\mathbb{F}_2^n$ *words* and elements in C *codewords*. Two interesting parameters of a code are its *redundancy* and its *noise-resiliency*. The redundancy is measured by the *rate* of the code, $\frac{\log|\mathcal{C}|}{n}$. The noise-resiliency is measured according to the model of noise.

In the errors model: A code $\mathcal{C}$ is $(\tau n, L)$ *list-decodable* if for every word $w \in \mathbb{F}_2^n$ there exist at most L codewords in the Hamming ball of radius τn around w.

In the erasures model: A code $\mathcal{C}$ is $(\tau n, L)$ *erasure list-decodable* if for every $z \in \mathbb{F}_2^{(1-\tau)n}$ and every set $T \subseteq [n]$ of size $(1-\tau)n$, the number of codewords that have z in the coordinates indexed by T is at most L.

If $\mathcal{C}$ is $(\tau n, L)$ list-decodable we can recover from τn errors in the following sense: Given a word $w \in \mathbb{F}_2^n$ that was obtained by corrupting at most τn entries of some codeword c, one can (perhaps non-efficiently) produce a small set of size L that necessarily contains c.

Similarly, if $\mathcal{C}$ is $(\tau n, L)$ erasure list-decodable we can recover from τn erasures in the following sense: Given a word $w \in \{0,1,?\}^n$ that was obtained by replacing at most τn entries of some codeword c with the erasure sign '?', one can (perhaps non-efficiently) produce a small set of size L that necessarily contains c.

A strong (k, ε) extractor with one output bit is roughly equivalent to a binary $(\frac{1-\varepsilon}{2}n, L = 2^k)$ list-decodable code [Tre01]. In the same spirit, Guruswami [Gur04a] observed that strong dispersers with one output bit can be used to construct erasure list-decodable codes. In this work we complement his argument with the converse statement, showing that erasure list-decodable codes are essentially *equivalent* to strong dispersers with one output bit. Specifically, $\mathsf{Disp} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$ is a strong (k, ε) disperser if and only if the code $\mathcal{C} : \{0,1\}^n \to \{0,1\}^{2^d}$ defined by $\mathcal{C}(x)_i = \mathsf{Disp}(x, i)$ is $((1-2\varepsilon)2^d, 2^k)$ erasure list-decodable.

As we can see, for both extractors and dispersers, the seed-length corresponds to the rate of the code, whereas the entropy loss corresponds to the list-size of the code. Thus, the gap between the seed-lengths of dispersers (which is $\log(\frac{1}{\varepsilon})$) and extractors (which is $2\log(\frac{1}{\varepsilon})$)

translates to a difference between rate ε in the erasures model compared with rate ε^2 in the errors model. Similarly, the gap between the entropy loss of dispersers (which is $\log\log(\frac{1}{\varepsilon})$) and extractors (which is $2\log(\frac{1}{\varepsilon})$) translates to a difference between list-size $\log(\frac{1}{\varepsilon})$ in the erasures model compared with list-size $\mathsf{poly}(\frac{1}{\varepsilon})$ in the errors model. Formally:

- Non-explicitly there exist binary codes having rate $\Omega(\varepsilon^2)$ that are $(\frac{1-\varepsilon}{2} \cdot n, \mathsf{poly}(\frac{1}{\varepsilon}))$ list-decodable and these parameters are tight.
- Non-explicitly there exist binary codes having rate $\Omega(\varepsilon)$ that are $((1-\varepsilon)n, O(\log \frac{1}{\varepsilon}))$ erasure list-decodable, and up to a constant multiplicative factor in the list-size these parameters are tight [Gur03].

Thus, erasure list-decodable codes can have quadratically better rate and exponentially smaller list-size than list-decodable codes. In fact, Guruswami proved that any *linear* erasure list-decodable codes must have $L = \Omega(1/\varepsilon)$ [Gur03], and so the exponential improvement (or any better than polynomial improvement) is necessarily only possible for non-linear constructions.

The state of affairs for *explicit* binary erasure list-decodable codes is similar to that of *explicit* dispersers. That is, only few explicit binary erasure list-decodable codes are known to have rate below $\Omega(\varepsilon^2)$ or list-size below $\Omega(1/\varepsilon)$. Guruswami and Indyk [GI02] gave a *probabilistic* polynomial-time algorithm that outputs with high probability an erasure list-decodable code of rate $\Omega\left(\frac{\varepsilon^2}{\log(1/\varepsilon)}\right)$ and optimal list-size (their construction can be explicitly derandomized when ε is constant). The natural open problem of obtaining erasure list-decodable codes having rate better than ε^2 was explicitly mentioned several times, e.g., in [GI02, Gur04a]. More concretely, in [Gur04b, Open Question 10.2], Guruswami posed the open problem of constructing efficient erasure list-decodable codes of rate ε^{2-a}.

Incorporating the above discussion with Theorem 6.1.1, we get the best explicit construction to date:

Theorem 6.1.2 (see also Theorem 6.5.8)**.** *For every constant $0 < \gamma < 1$ and $\varepsilon = n^{-\Omega(1)}$ there exists an explicit $((1-\varepsilon)\bar{n}, L = \log^{O(1)} \frac{1}{\varepsilon})$ erasure list-decodable code $\mathcal{C}\colon \{0,1\}^n \to \{0,1\}^{\bar{n}}$ of rate $\varepsilon^{1+\gamma}$, where the asymptotic notation hides constants that may depend on γ.*

Thus, Theorem 6.1.2 makes progress on resolving Guruswami's question for the interesting regime of polynomially-small ε. We stress that the codes we present are explicit in the sense that they have explicit encoding, but we do not know whether the codes we construct admit *efficient* erasure list-decoding algorithms. We also mention that the list-size $\mathsf{polylog}(\frac{1}{\varepsilon})$ achieved by our code is exponentially smaller than the best possible list-size by any *linear* code.

6.1.2 Two-source extractors and strong dispersers

Often, the two-source extractor terminology is more expressive than the extractor notation, as we explain now. Suppose $\mathsf{Ext}\colon \{0,1\}^k \times \{0,1\}^d \to \{0,1\}$ is a strong (k,ε) extractor. Fix an (n,k) source X, and let ε_i be the distance of the distribution $\mathsf{Ext}(X,i)$ from uniform. By the extractor definition we know that $\mathbb{E}[\varepsilon_i] \leq \varepsilon$. However, the extractor definition does not

distinguish between the case where the ε error occurs because all seeds $y \in \mathrm{Supp}(Y)$ have the same error ε, and the case where ε fraction of the seeds have constant error and the rest have none. The situation is different with two-source extractors. Roughly speaking, in an $((n,k),(d,d'),\varepsilon)$ two-source extractor, there are at most $2^{d'-d}$ bad seeds y with distance $\varepsilon_y \geq \varepsilon$. Thus, the two-source extractor notation allows separating the fraction of bad seeds from the quality of good seeds.

We would like to explicitly construct a strong (k,ε) disperser $\mathsf{Disp} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with parameters better than those of (k,ε) extractors. Thus, on the one hand, for almost every seed y, $\mathsf{Disp}(X,y)$ covers almost all of $\{0,1\}^m$, and, on the other hand, Disp is not a strong extractor, so for almost every seed y, $\mathsf{Disp}(X,y)$ is *far* from uniform. How can this happen?

The situation becomes clearer if we look at strong dispersers with only one additional output bit, i.e., when $m = 1$. As $\mathsf{Disp}(X,y)$ is supported on one bit, for almost every seed y, $\mathrm{Supp}(\mathsf{Disp}(X,y)) = \{0,1\}$. Yet, it is possible (even necessary, since Disp is not an extractor) that for many seeds y, $\mathsf{Disp}(X,y)$ is ε_0 away from uniform for some constant $\varepsilon_0 \gg \varepsilon > 0.$, e.g., when $\mathsf{Disp}(X,y)$ has much more weight on 0 than on 1.

One clean way of capturing this is by using the two-source extractor terminology. We are looking for a two-source extractor $\mathsf{2Ext}$ where almost all seeds (except for ε fraction) are "good" in the sense that y is good if $\mathsf{2Ext}(X,y)$ covers both 0 and 1. Roughly speaking, this amounts to an explicit construction of an $((n,k),(d,d'),\varepsilon_0)$ two-source extractor having $\varepsilon = 2^{d'-d}$ and any non-trivial error $\varepsilon_0 < 1$. As we already mentioned, two-source extractors with arbitrary $\varepsilon_0 < 1$ are also called bipartite Ramsey graphs (see also Claim 6.5.11).

Explicitly constructing two-source extractors (and Ramsey graphs) is a long standing and important challenge and in previous chapters we gave the state-of-the-art constructions, in the *balanced* regime. However, using such extractors gives dispersers with suboptimal entropy loss and long seed, or, equivalently, erasure list-decodable codes with large list-size and low rate.

Another natural two-source extractor is Raz's two-source extractor [Raz05]. Raz's extractor is an $((n,k),(d = O(\log \frac{n}{\varepsilon}),d'),\varepsilon_{\mathsf{Raz}})$ two-source extractor that has an *unbalanced* entropy requirement; the first source is long and very weak (k can be as small as, roughly, $\log\log \frac{n}{\varepsilon_{\mathsf{Raz}}}$), the second source is short and somewhat dense with $d' \geq \delta d$, for any constant $\delta > \frac{1}{2}$. The fact that k can be very small corresponds to a disperser with small entropy loss, which is good for us. Moreover, d is small, which is again what we want because the length of the corresponding erasure list-decodable code is 2^d. The error $\varepsilon_{\mathsf{Raz}}$ of Raz's extractor is exponentially-small in $\min\{k,d'\}$ which is much better than the mere non-trivial error that we need. However, the second source must be relatively dense, satisfying $\frac{d'}{d} \geq \frac{1}{2}$. This implies that the error ε of the disperser is given by $2^{-d+d'}$ and as a consequence $d \geq 2\log(\frac{1}{\varepsilon})$.

In this chapter we show how to explicitly construct the necessary two-source extractor. We prove:

Theorem 6.1.3 (see also Theorem 6.4.1)**.** *For every two constants $\delta, \varepsilon_0 > 0$ and every $k \geq \Omega(\log\log n)$ there exists an explicit $((n,k),(d,\delta d),\varepsilon_0)$ two-source extractor $\mathsf{2Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$ with $d = O(\log n)$.*

Theorem 6.1.3 is interesting on its own right. The entropy requirement in both sources is optimal up to constant factors, as both sources have entropy which is logarithmic in

the length of the other source. This property is also true for Raz's extractor. On the negative side, Theorem 6.1.3 has a large error ε_0, whereas Raz's extractor has a very small error. On the positive side, Raz's extractor works only when $d' = \delta d > 0.5d$ whereas Theorem 6.1.3 works with $d' = \delta d$ for any $\delta > 0$, and it is this feature that gives a disperser construction with parameters better than those possible for extractors. Having Theorem 6.1.3 immediately gives the strong one output bit disperser and the non-linear near-optimal erasure list-decodable code discussed above.

We also obtain a variant of Theorem 6.1.3 that gives a new construction of balanced two-source extractors.

Theorem 6.1.4. *For every two constants $\delta, \varepsilon_0 > 0$ and every $k \geq \Omega(\log n)$ there exists an explicit $((n,k),(n,\delta n),\varepsilon_0)$ two-source extractor* $\mathsf{2Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$.

We see that one source has a minimal entropy requirement of $O(\log n)$ while the other has arbitrarily small constant entropy rate. Again, this improves upon [Raz05] in terms of entropy requirement but is worse in terms of error. Theorem 6.1.4 is also incomparable to [Li18] as there, both sources require min-entropy at least $O(\log n \frac{\log\log n}{\log\log\log n})$.

Both Theorem 6.1.3 and Theorem 6.1.4 follow directly from Theorem 6.4.1.

6.1.3 The two-source extractor construction

We now give an informal presentation of the two-source extractor construction. We try to keep the discussion intuitive, and for that we omit (or ignore) some technical details.

The input to the $((n,k),(d,\delta d),\varepsilon_0)$ two-source extractor is an (n,k) source X and a $(d,\delta d)$ source Y, for some $0 < \delta < \frac{1}{2}$. We would like to do the following:

1. Increase the entropy rate of Y from δ to, say, 0.7. For that, we use a constant-error *condenser*. We cannot do it deterministically (because the condenser needs a uniformly random seed) and we still want to keep X fresh. Therefore, we apply the condenser on Y and every possible seed, letting the output of this procedure be a table Y' in which each row corresponds to an application with a different seed. The table Y' has the guarantee that most of the rows of Y' are close to having entropy rate 0.7.

2. Next, we would like to transform the dense rows of Y' to uniformly random strings. For that, we use Raz's extractor with the first source X and the rows of Y' as (independent) seeds. Call the resulting table Y'' and note that it is a function of both X and Y. Also note that although it is now guaranteed that a constant fraction of the rows of Y'' are uniform (Raz's extractor works with entropy rate above half), it is *not* guaranteed (and also not true) that the rows of Y'' are independent of each other.

3. Now we wish to break the dependence between the rows of Y'' so that (ideally) every t of them are uniform and independent (think of t as being poly-logarithmic in the number of rows of Y''). For that, we use a *correlation-breaker* that outputs one bit. The correlation-breaker requires two independent sources, which we do *not* have. Instead, we apply it on Y and Y''. Call the output table Y'''. We shall prove that with high probability, Y''' has many good rows and every t good rows of Y''' are *very* close to being uniform and independent.

4. Finally, we apply a *resilient function* f on the bits of Y'''. The output of our construction is the function's output $f(Y''')$.

   The property that we want from f is that it is nearly balanced and that its output cannot be heavily influenced by any small set of bad bits (the bad rows of Y'''). We need these properties to hold not only when the "good" bits are perfectly uniform and independent, but also under weaker conditions (e.g., that the good players are t-wise independent).

Our construction shares steps that are similar to Cohen's construction [Coh16a] of three-source extractors. The vital difference is that in [Coh16a], a third source is used to achieve complete independence between the rows of a table and then a simple parity can be applied, even if only one row is close to uniform. Here, we only use *two* sources. The use of only two sources raises several delicate issues:

- First, there is the issue of lack of independence between the source Y and the seed Y'' in item (3) of the construction. To overcome this, we show a conditioning under which Y'' is still good, Y is independent of Y'' and even after the conditioning the two sources have enough min-entropy. In recent years, such prevalent conditioning methods were very successful in constructing an abundance of primitives (e.g., correlation breakers, independence-preserving mergers and non-malleable extractors).

- Next, there is a delicate issue with the errors. The error $\varepsilon_{\mathsf{Cond}}$ of the condenser is high (think of it as a constant). In a naive analysis we would argue that each t good rows are $\varepsilon' > \varepsilon_{\mathsf{Cond}}$ close to uniform, and therefore the whole table Y''' is $A^t\varepsilon'$-close to a table where the good rows are perfectly t-wise independent, where A is the number of rows in the table Y'''. However, such an approach is doomed to fail, as necessarily $A\varepsilon_{\mathsf{Cond}} > 1$.

  The solution for this problem is at the heart of the argument. We observe that some of the errors in the construction depend on A, the number of rows in the table, while others depend on the *row length*. In the construction we make sure that A is small (think of it as a fixed constant) while the row length is unbounded (and, e.g., grows to infinity as n grows to infinity). Thus, we have a natural separation between *large* errors that depend on the number of rows A, and *small* errors that depend on the row length.

  The condenser of step (1) and the resilient function of step (4) incur large errors. Raz's extractor (step (2)) and the correlation breaker with advice (step (3)) incur small errors that are exponentially-small in the row length. We show that with some constant probability we succeed in step (1), and that once we have succeed, the errors δ in steps (2) and (3) are so small that $A^t\delta$ is still small, hence Y''' is close to a table with t-wise independent good players, and so the resilient function in step (4) works (and incurs another constant error). Thus, while the failure probability is high, when we succeed we are exponentially-close to uniform.

- Finally, the argument used in the last bullet raises a difficulty treating the set of good rows. Specifically, in [CZ16], the set of good rows is a function of one of the sources.

In our analysis the set of good rows is not just a function of the *sources* X and Y, but also depends on the specific *sample* $y \sim Y$.

6.1.4 Non-strong dispersers

Strong dispersers are the the focal point of this work. One may wonder why we insist on the strongness property, and whether the problem becomes easier when the strongness property is dropped.

- The answer to the first question is that the strongness property is essential. The equivalence between erasure list-decodable codes and dispersers requires the dispersers to be strong (see Lemma 6.5.6, and also notice the correspondence between code coordinates and seeds). Similarly, the connection to Ramsey graphs also requires the disperser to be strong, as already observed by Gradwohl et al. [GKRTS05]. [GKRTS05] constructed dispersers that are strong in almost all of the seed, but not strong in some part of the seed, and this drawback is severe enough that none of the applications go through.

- The answer to the second question is that it is easier to construct non-strong dispersers with good parameters. We show that it is possible to output more bits from the source at the expense of being strong in only most of the bits (we are non-strong in only $O(1)$ bits of the seed). We prove:

  Theorem 6.1.5. *For every constant $0 < \gamma < 1$ and $\varepsilon = n^{-\Omega(1)}$ there exists an explicit (k, ε) disperser $\mathsf{Disp}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = (1+\gamma)\log(\frac{1}{\varepsilon})$, $k \geq \Omega(\log\log\frac{1}{\varepsilon})$ and $m = d + \Omega(k)$, where the constant in the $O(\cdot)$ notation may depend on γ. The disperser is strong in $d - O(1)$ bits of the seed.*

  We sketch a proof of the above theorem in Section 6.5.2.

6.1.5 Organization

The rest of the chapter is organized as follows. Section 6.2 covers the preliminaries and notations we use and did not appear in Chapter 2. Section 6.3 describes the *constant degree* condenser that is used in step (1). Following the above discussion, it is important for us that A, the number of rows in the table, and equivalently the seed-length of the condenser, is a constant independent of the row length. In that section we show one can combine existing constructions of somewhere-random condensers and mergers to achieve that. Next, in Section 6.4, we describe and analyze the new unbalanced two-source extractor. In Section 6.5 we use the new two-source extractor to obtain near-optimal strong seeded dispersers, erasure list-decodable codes and unbalanced Ramsey graphs. We conclude with a few open problems in Section 6.6.

6.2 Preliminaries

6.2.1 Strong seeded condensers and two-source extractors

We prove that every seeded condenser satisfies some strongness property.

Lemma 6.2.1. *Suppose* $\mathsf{Cond}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* $(k \to k'+d, \varepsilon_{\mathsf{Cond}})$ *condenser. Let* X *be an* (n,k) *source. Let* ε_i *be the minimal distance of* $\mathsf{Cond}(X,i)$ *to an* (m,k') *source. Then,* $\mathbb{E}_{i\in\{0,1\}^d}[\varepsilon_i] \leq \varepsilon_{\mathsf{Cond}}$.

Proof. Fix an (n,k) source X. For $i \in \{0,1\}^d$, let $H_i \subseteq \{0,1\}^m$ be the set of elements $w \in \{0,1\}^m$ such that $\Pr_{x\in X}[\mathsf{Cond}(x,i) = w] \geq 2^{-k'}$. The distance of $\mathsf{Cond}(X,i)$ from a k'-source is $\varepsilon_i = \Pr_{x\in X}[C(x,i) \in H_i] - 2^{-k'}|H_i|$. Let $H = \bigcup_{i\in\{0,1\}^d} H_i$. Then,

- For every $w \in H$, $\Pr_{x\in X, i\in\{0,1\}^d}[\mathsf{Cond}(x,i) = w] \geq 2^{-d}2^{-k'} = 2^{-(k'+d)}$, and,
- it holds that

$$\begin{aligned}
\varepsilon_{\mathsf{Cond}} &\geq \Pr_{x\in X, i\in\{0,1\}^d}[\mathsf{Cond}(x,i) \in H] - |H|2^{-(k'+d)} \\
&= \sum_{i\in\{0,1\}^d} 2^{-d} \Pr_x[\mathsf{Cond}(x,i) \in H] - |H|2^{-(k'+d)} \\
&\geq \sum_{i\in\{0,1\}^d} 2^{-d} \Pr_x[\mathsf{Cond}(x,i) \in H_i] - 2^{-(k'+d)} \sum_{i\in\{0,1\}^d} |H_i| \\
&= \sum_{i\in\{0,1\}^d} 2^{-d} \left(\Pr_x[\mathsf{Cond}(x,i) \in H_i] - 2^{-k'}|H_i|\right) \\
&= \sum_{i\in\{0,1\}^d} 2^{-d}\varepsilon_i \;=\; \mathbb{E}_{i\in\{0,1\}^d}[\varepsilon_i].
\end{aligned}$$

□

Definition 6.2.2. *We say that an* $((n_1,k_1),(n_2,k_2),\varepsilon)$ *two-source extractor* $\mathsf{2Ext}\colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ *is* strong *if*

$$(\mathsf{2Ext}(X_1,X_2), X_1) \approx_\varepsilon (U_m, X_1)$$

and

$$(\mathsf{2Ext}(X_1,X_2), X_2) \approx_\varepsilon (U_m, X_2).$$

◊

In our construction, we will use Raz's two-source extractor.

Theorem 6.2.3 ([Raz05])**.** *For every constant* $\delta_{\mathsf{Raz}} > 1/2$ *there exist constants* $c_1 = c_1(\delta_{\mathsf{Raz}}), c_2 = c_2(\delta_{\mathsf{Raz}}) > 1$ *such that for every integers* n_1, k_1, n_2, k_2 *satisfying*

- $k_1 \geq c_1 \log n_2$,
- $k_2 \geq c_2 \log n_1$,

there exists an explicit strong $((n_1,k_1),(n_2,k_2 = \delta_{\mathsf{Raz}} n_2),\varepsilon_{\mathsf{Raz}})$ *two-source extractor*

$$\mathsf{Raz}\colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$$

with $m = \Omega(\min\{k_1,k_2\})$ *and* $\varepsilon_{\mathsf{Raz}} = 2^{-\Omega(m)}$*, where the constants hiding in the asymptotic notation may depend on* δ_{Raz}.

Claim 6.2.4. *Suppose*

$$\mathsf{2Ext}\colon \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$$

is a strong $((n_1,k_1),(n_2,k_2),\varepsilon)$ two-source extractor. Let X be an (n,k_1) source. Call an element $y \in \{0,1\}^{n_2}$ is bad *if $|\mathsf{2Ext}(X,y) - U_m| > \varepsilon$, and let BY denote the set of all bad elements. Then, $|BY| < 2^{k_2}$.*

Proof. Assume towards contradiction that $|BY| \geq 2^{k_2}$ and let Y be the uniform distribution over the set BY. Then, $H_\infty(Y) \geq k_2$ and so $(\mathsf{2Ext}(X,Y),Y) \approx_\varepsilon (U_m,Y)$ which implies that

$$\frac{1}{|BY|}\sum_{y\in BZ} |\mathsf{2Ext}(X,y) - U_m| \;\leq\; \varepsilon.$$

However, $|\mathsf{2Ext}(X,y) - U_m| > \varepsilon$ for every $y \in BY$, in contradiction. □

6.2.2 Correlation breakers with advice

A correlation-breaker with advice is a function $\mathsf{CBA}\colon \{0,1\}^n \times \{0,1\}^\ell \times [A] \to \{0,1\}^m$ where we think of the first input as a weak source, the second as an independent short seed and the last as an advice string. Roughly speaking, applying CBA on t possibly correlated seeds with t distinct advice strings results in independent random variables. For example, $\mathsf{CBA}(X,Y,\alpha)$ is (nearly) independent of $\mathsf{CBA}(X,Y,\alpha')$ for any $\alpha \neq \alpha'$. Formally,

Definition 6.2.5. *A function $\mathsf{CBA}\colon \{0,1\}^n \times \{0,1\}^\ell \times [A] \to \{0,1\}^m$ is a $(t,k,\varepsilon_{\mathsf{CBA}})$ correlation-breaker with advice if the following holds. If Y is a distribution over $\{0,1\}^n$, $Z = (Z_1,\ldots,Z_t)$ is a distribution on $(\{0,1\}^\ell)^t$, $\mathcal{H}$ is a random variable and $\delta > 0$ satisfying:*

- *Y and Z are independent, conditioned on $\mathcal{H}$,*
- *$\tilde{H}_\infty(Y|H) \geq k + \log(1/\varepsilon_{\mathsf{CBA}})$,*
- *$(Z_1,\mathcal{H}) \approx_\delta (U_\ell,\mathcal{H})$, and,*
- *$\alpha_1,\ldots,\alpha_t \in [A]$ are distinct strings.*

Then,

$$\left(\mathsf{CBA}(Y,Z_1,\alpha_1),(\mathsf{CBA}(Y,Z_i,\alpha_i))_{i=2}^t,\mathcal{H}\right) \approx_{\delta+2\varepsilon_{\mathsf{CBA}}} \left(U_m,(\mathsf{CBA}(Y,Z_i,\alpha_i))_{i=2}^t,\mathcal{H}\right).$$

♢

We use the following result:

Theorem 6.2.6 ([Coh16c])**.** *There exists a constant $c_{\mathsf{CBA}} \geq 1$ such that the following holds. Let n,a be integers and $\varepsilon_{\mathsf{CBA}} > 0$. Then, there exists an explicit $(t,k_{\mathsf{CBA}},\varepsilon_{\mathsf{CBA}})$ correlation-breaker with advice*

$$\mathsf{CBA}\colon \{0,1\}^n \times \{0,1\}^\ell \times [A] \to \{0,1\}$$

with $\ell = c_{\mathsf{CBA}} \cdot at \cdot \log\frac{n}{\varepsilon_{\mathsf{CBA}}}$ and $k_{\mathsf{CBA}} \geq \ell$.

In our setting, the number of rows A is a constant independent of n. For this reason we work with a "basic" correlation-breaker, where there is no attempt to optimize the dependence of ℓ on a. This gives a seed-length which is optimal up to constant multiplicative factors.

6.3 Constant-Degree Condensers

In this section we prove:

Theorem 6.3.1. *For every constant $0 < \delta_1 < \delta_2 = 0.7$, every $s \geq s_0(\delta_1)$ and every integer n_1 and $\varepsilon_{\mathsf{Cond}} \geq 2^{-\Omega(n_1)}$ there exists an explicit rate $(\delta_1 \to \delta_2, \varepsilon_{\mathsf{Cond}})$ condenser*

$$\mathsf{Cond}\colon \{0,1\}^{n_1} \times \{0,1\}^{d} \to \{0,1\}^{n_2}$$

with $n_2 = (\frac{2}{3})^s n_1$ and $d = 4c_{\mathsf{DKSS}}\left(s + \log \frac{1}{\varepsilon_{\mathsf{Cond}}}\right)$, where c_{DKSS} is the constant from Theorem 2.3.15. Note that d is independent of n_1.

Note that, in particular, for every $\delta_1 > 0$ there exists an explicit $(\delta_1 \to \delta_2 = 0.7, \varepsilon_{\mathsf{Cond}})$ condenser $C\colon \{0,1\}^{n_1} \times \{0,1\}^{d} \to \{0,1\}^{n_2}$ with $n_2 = \Omega(n_1)$ and $d = O(\log \frac{1}{\varepsilon_{\mathsf{Cond}}})$. However, we will need the more precise version that appears in Theorem 6.3.1.

The proof goes through *somewhere-random condensers*, which we saw in Section 2.3.3. Recall that a function

$$\mathsf{SRC}\colon \{0,1\}^{n} \to (\{0,1\}^{m})^{A}$$

is a seedless $(k \to k', \varepsilon)$ s.r. condenser if for every (n,k) source X it holds that

$$\mathsf{SRC}(X) = \mathsf{SRC}(X,1) \circ \ldots \circ \mathsf{SRC}(X,A)$$

is ε-close to a k' s.r. source. If $k = \delta n$ and $k' = \delta m$ we say SRC is $(\delta \to \delta', \varepsilon)$ s.r. condenser.

We may take a condenser $\mathsf{Cond}\colon \{0,1\}^{n} \times \{0,1\}^{d} \to \{0,1\}^{m}$ and expand it to a table with the outputs of all possible seeds, i.e., define $S\colon \{0,1\}^{n} \to (\{0,1\}^{m})^{D}$, with $D = 2^d$, where $S(x)_i = \mathsf{Cond}(x,i)$. The condenser property guarantees that for every k-source X, *most* rows in the table are close to having k' min-entropy. In contrast, a s.r. condenser is a weaker object, because it only guarantees that *one* row has k' entropy (or more precisely that we are in a convex combination of such cases).

The major question we consider now is the dependence of the degree (2^d for condensers and A for s.r. condensers) on n, m, k, k' and ε. We focus on the case where $m = \Omega(n)$, $k = \delta n$, $k' = \delta' m$ and $\delta < \delta'$ are constants. A-priori, we could have expected the degree to depend on n and ε, as is indeed the case when m might be arbitrarily small. However, remarkably, things are drastically different when $m = \Omega(n)$. In this case both condensers and s.r. condensers may be of degree that is independent of n and this will be crucial for us. If we consider the dependence on the error, then s.r. condensers may have exponentially-small error and constant D, whereas the degree of a condenser is at least $d \geq \log(\frac{1}{\varepsilon})$. Remarkably, all of that can be explicitly achieved, as we now explain.

The basic building block we use is the following beautiful result of Zuckerman, already surveyed in Section 2.3.3.

Theorem 6.3.2 ([Zuc07], Theorem 2.3.13 rephrased)**.** *For every constant $0 < c < 1$ there exists a constant $\alpha = \alpha(c)$ such that for every constant $\delta \leq c$ and integer n there exists an explicit function $\mathsf{Z}\colon \{0,1\}^{n} \to (\{0,1\}^{\frac{2}{3}n})^{2}$ that is a rate $(\delta \to (1+\alpha)\delta, \varepsilon)$ s.r. condenser with $\varepsilon = 2^{-\Omega(\alpha\delta n)}$.*

Somewhere-random condensers can be easily composed. Barak et al. [BKS+10] showed that if

$$\mathsf{SRC}_1\colon \{0,1\}^{n_1} \to (\{0,1\}^{n_2})^{\ell_1}$$

is a rate $(\delta_1 \to \delta_2, \varepsilon_1)$ s.r. condenser and

$$\mathsf{SRC}_2\colon \{0,1\}^{n_2} \to (\{0,1\}^{n_3})^{\ell_2}$$

is a rate $(\delta_2 \to \delta_3, \varepsilon_2)$ s.r. condenser then

$$\mathsf{SRC}_2 \circ \mathsf{SRC}_1\colon \{0,1\}^{n_1} \to (\{0,1\}^{n_3})^{\ell_1 \cdot \ell_2}$$

defined by $\mathsf{SRC}_2 \circ \mathsf{SRC}_1(x)_{(i_1,i_2)} = \mathsf{SRC}_2(\mathsf{SRC}_1(x)_{i_1})_{i_2}$ is a rate $(\delta_1 \to \delta_3, \varepsilon_1 + \varepsilon_2)$ s.r. condenser.

Composing the s.r condenser Z of Theorem 6.3.2 with itself s times we get an explicit function

$$\mathsf{SRC}\colon \{0,1\}^{n} \to (\{0,1\}^{m})^{D}$$

with $D = 2^s$ and $m = (\frac{2}{3})^s n$ that is a rate $(\delta \to \delta', \varepsilon)$ s.r. condenser with

$$\varepsilon = \sum_{i=1}^{s} 2^{-\Omega((1+\alpha)^i \delta (\frac{2}{3})^i n)} = 2^{-\Omega(m)}$$

and $\delta' \geq (1 + \alpha(\delta'))^s \delta$. Therefore:

Lemma 6.3.3. *For every constants $0 < \delta_1 < \delta_2 < 1$ there exists a constant $s = s(\delta_1, \delta_2)$ and an explicit function $\mathsf{SRC}\colon \{0,1\}^{n_1} \to (\{0,1\}^{n_2})^D$ that is a $(\delta_1 \to \delta_2, \varepsilon)$ s.r. condenser with $D = 2^s$, $n_2 = (\frac{2}{3})^s n_1$ and $\varepsilon = 2^{-\Omega(n_2)}$. Note that D is independent of n and ε.*

Right now, if X is a k-source, the table $\mathsf{SRC}(X)$ has D rows, and, roughly speaking, the guarantee is that one of these rows has density δ'. We want to change this to get a condenser, i.e., we are willing to invest a short seed (that is independent of n) and we want to get *one* output which is close to uniform. (Alternatively, we can write the condenser as a table with one row per seed, the number of rows is independent of n and most rows are close to uniform.) This is exactly what a *merger* does (see Section 2.3.4) and applying the merger of Theorem 2.3.15 with $\beta = \frac{1}{4}$ on the s.r. condenser of Lemma 6.3.3 (with δ_2 close to 1) gives Theorem 6.3.1

6.4 The Unbalanced Two-Source Extractor Construction

The main result of this section is the following two-source extractor.

Theorem 6.4.1. *For every integer n and two constants $\delta_0, \varepsilon_0 > 0$ there exists a constant c such that for all integers $d \geq c \log n$ and $k \geq c \log d$ there exists an explicit $((n,k),(d,\delta_0 d),\varepsilon_0)$ two-source extractor $\mathsf{2Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$.*

The extractor in the above theorem has constant error, and works when:

1. Each source's entropy is in the order of the logarithm of the length of the other source.
2. The shorter source, of length d, has an arbitrarily small constant density δ_0.

We think of n and $d = d(n)$ as growing parameters while ε_0 and δ_0 are constants. We use asymptotic notations (such as $\Omega(\cdot)$) to hide constants that are independent of n and d (but may depend on ε_0 and δ_0).

6.4.1 The construction

Recall that ε_0 is the target error of the extractor $\mathsf{2Ext}$. The input to $\mathsf{2Ext}$ is a pair (x, y) where x is drawn from an (n, k) source X, and y is drawn from an independent $(d, \delta_0 d)$ source Y. Our problem is that the y comes from a $\delta_0 d$-source for some $\delta_0 < \frac{1}{2}$. To overcome this, we do the following:

- We apply the condenser of Theorem 6.3.1 on y to get a table y' that is 1-wise 0.7-dense. Notice that the output of this step is a table rather than a single output.
- We apply Raz's extractor (Theorem 6.2.3) on the table and the input x from the other source to convert the table y' to another table y'' that is 1-wise uniform.
- We apply the t correlation-breaker with advice of Theorem 6.2.6 on y, using the table y'' as the seed, to get a table y''' that is t-wise uniform.
- Finally, we apply the resilient function f of Theorem 2.1.15 on the table y''' to collapse the many rows of the table to a single, close to uniform, output.

Formally, these steps work as follows:

Condense the short source: We are given $\delta_0 < \frac{1}{2}$. Set $\delta' = 0.69$ and $\delta_2 = 0.7$.

By Theorem 6.3.1 there exists a constant $s_0 = s_0(\delta_0)$ such that for every $s \geq s_0$ there exists an explicit

$$\mathsf{Cond}\colon \{0,1\}^d \times \{0,1\}^a \to \{0,1\}^{d'}$$

that is a rate $(\delta_0 \to \delta_2 = 0.7, \varepsilon_{\mathsf{Cond}})$ condenser with $a = 4c_{\mathsf{DKSS}}(s + \log \frac{1}{\varepsilon_{\mathsf{Cond}}})$ and $d' = (\frac{2}{3})^s d$.

We set

$$\gamma = \frac{1}{2^5 c_{\mathsf{DKSS}}},$$

and this also fixes c_γ as in Theorem 2.1.15. Notice that γ and c_γ are fixed constants independent of all other parameters in our system.

Now, choose $\varepsilon_{\mathsf{Cond}}$ so that

$$\left(\frac{1}{\varepsilon_{\mathsf{Cond}}}\right)^{\log(3/2)} \geq \frac{4}{\delta_0} 2^{12} c_\gamma {c_{\mathsf{DKSS}}}^4 \log^4 \frac{1}{\varepsilon_{\mathsf{Cond}}}, \tag{6.2}$$

and also so that $\varepsilon_{\mathsf{Cond}} \leq \xi(\varepsilon_0, \delta_0)$, where

$$\xi(\varepsilon_0, \delta_0) = \min\left\{2^{-s_0}, \left(\frac{\varepsilon_0}{8}\right)^2, \left(\frac{\varepsilon_0}{5}\right)^{c_\gamma}, \left(\frac{\varepsilon_0}{5c_\gamma}\right)^{1/\gamma}\right\}. \tag{6.3}$$

Given $\varepsilon_{\mathsf{Cond}}$, we set

$$s = \log\frac{1}{\varepsilon_{\mathsf{Cond}}} \geq s_0,$$

giving $a = 8c_{\mathsf{DKSS}} \log\frac{1}{\varepsilon_{\mathsf{Cond}}}$. Note that the degree of the condenser, $A = 2^a$, satisfies

$$\sqrt{\varepsilon_{\mathsf{Cond}}}A \;=\; 2^{-\frac{1}{2}\log\frac{1}{\varepsilon_{\mathsf{Cond}}}+a} \;=\; 2^{-\frac{a}{2^4 c_{\mathsf{DKSS}}}+a} \;=\; A^{1-2\gamma}.$$

Observe that $s \geq s_0$ and that $d' = \Omega(d)$. Also, notice that

$$(\delta_2 - \delta')d' = d'/100 \geq a = \log A$$

for large enough d. Thus, Cond is a $(\delta d \to \log(A) + \delta' d', \varepsilon_{\mathsf{Cond}})$ condenser.

Define an $A \times d'$ table Y' where

$$Y'_i = C(Y, i) \in \{0,1\}^{d'}$$

for $i = 1, \ldots, A$.

1-wise uniformity: Let c_1, c_2 be the constants from Theorem 6.2.3 for $\delta_{\mathsf{Raz}} = 0.6$.

Notice that $\delta_{\mathsf{Raz}} d' = \Omega(d') = \Omega(d)$. Therefore, for a constant c large enough, $d \geq c \log n$ is large enough so that $\delta_{\mathsf{Raz}} d' \geq c_2 \log n$. We can, in particular, choose c such that in addition $c \geq c_1$. Recalling that $k \geq c \log d$, we have $k \geq c_1 \log d'$. By Theorem 6.2.3, there exists an explicit function

$$\mathsf{Raz}\colon \{0,1\}^n \times \{0,1\}^{d'} \to \{0,1\}^{d''}$$

that is a strong $((n,k),(d', \delta_{\mathsf{Raz}} d'), \varepsilon_{\mathsf{Raz}} = 2^{-\Omega(d'')})$ two-source extractor with $d'' = \Omega(\min\{k, \delta_{\mathsf{Raz}} d'\}) = \Omega(k)$.[1] Define an $A \times d''$ table Y'' where

$$Y''_i = \mathsf{Raz}(X, Y'_i)$$

for $i = 1, \ldots, A$.

t-wise uniformity: Let $k_{\mathsf{CBA}} = \frac{\delta_0 d}{8}$ and $\varepsilon_{\mathsf{CBA}} = \frac{1}{d}$. Set

$$t = \frac{\delta_0}{4}\left(\frac{3}{2}\right)^s.$$

[1] Although $k \geq c \log d$ we can always assume without loss of generality that $k = c \log d$ and so $k \leq \delta_{\mathsf{Raz}} d' = \Omega(d)$.

Notice that for a large enough constant c we have

$$d'' = \Omega(k) = \Omega(c \log d) \geq c_{\mathsf{CBA}} at \log \frac{d}{\varepsilon_{\mathsf{CBA}}},$$

where the latter is the seed-length required by the correlation-breaker from Theorem 6.2.6. Also, $k_{\mathsf{CBA}} = \frac{\delta_0 d}{8} \geq d''$ for large enough d, as $d'' = \Omega(k) = \Omega(\log d)$. Hence, by Theorem 6.2.6 there exists an explicit function

$$\mathsf{CBA}\colon \{0,1\}^d \times \{0,1\}^{d''} \to \{0,1\}$$

that is a $(t, k_{\mathsf{CBA}}, \varepsilon_{\mathsf{CBA}})$ correlation-breaker with advice.

Define an $A \times 1$ table Y''' where

$$Y'''_i = \mathsf{CBA}(Y, Y''_i, i)$$

for $i = 1, \ldots, A$.

Keep in mind that the entropy in Y suffices for CBA since $H_\infty(Y) = 8k_{\mathsf{CBA}}$.

Collapse: Take $f : \{0,1\}^A \to \{0,1\}$ to be the $(q = A^{1-2\gamma}, t, \varepsilon_f = c_\gamma A^{-\gamma})$ resilient function of Theorem 2.1.15 and output $f(y'''_1, \ldots, y'''_A)$.

6.4.2 Two subtleties

As mentioned in the introduction, there are several delicate issues in the analysis:

1. Circular dependence: Y'' depends on both X and Y, and is used as a seed in the application of the correlation-breaker with advice on Y.
2. We need Y''' to be close to a perfect t-wise independent table, while the correlation-breaker with advice only guarantees that every t good rows are close to uniform. To bridge the gap we need the error to be at least polynomially-small in the number of rows, but some of the steps incur a large constant error.

To overcome the first issue we show a conditioning under which Y'' is still good, Y is independent of Y'' and even after the conditioning the two sources have enough min-entropy.

To overcome the second issue we distinguish between large errors that depend on the number of rows A, and small errors that depend on the row length (see Section 6.1.3). In particular, the errors are of three types:

- The probability p_1 that a value we condition upon is bad. This error is incurred by the condenser and is high (think of it as being a constant).
- We show that when we condition on a good value, every t good rows in Y''' are p_2-close to uniform. We then claim that Y''' as a table is $A^t p_2$-close to a table where the good rows are truly t-wise independent (where A is the number of rows in the table Y'''). The error p_2 is incurred by Raz's extractor and by the correlation-breaker, and can be made very small if we deal with a source X having enough min-entropy. We make p_2 small enough so that $A^t p_2$ is also small.

- A third error p_3 is incurred by the resilient function f. This error is large, say, a constant, and we are fine with that.

Note that we cannot just accumulate all errors as $A^t p_1$ is way larger than 1. Instead, we argue that with a constant probability $1 - p_1$, we get extremely close to perfect behavior, and then we get such a small error p_2 so that $A^t p_2$ is also small.

6.4.3 The analysis

Proof of Theorem 6.4.1. Fix an (n, k) source X and an independent $(d, \delta d)$ source Y. We decompose the proof into three parts:

- In the first part we prove that very often (except for a small constant probability) the table Y'' contains many rows that are marginally close to uniform.
- Next, we prove that every set of t rows $\{i, j_1, \ldots, j_{t-1}\}$ in Y''' are *product* in the sense that if i is a good row (intuitively meaning that Y''_i is close to uniform) and $j_1, \ldots, j_{t-1}$ are $t-1$ other rows, then in Y''', Y'''_i is close to uniform and *independent* of $Y'''_{j_1}, \ldots, Y'''_{j_{t-1}}$. This part involves applying a correlation-breaker with advice on Y and Y''. In order to ensure that Y and Y'' are independent, we condition on the values of Y' in the t rows $\{i, j_1, \ldots, j_{t-1}\}$.
- Together, except for a small constant probability, there are many good rows, and every t rows of Y''' are product, hence the table Y''' is close to a (q, t) non-oblivious bit-fixing source, where every good row is a good bit in the bit-fixing source. Hence, $f(Y''')$ is close to uniform.

Part 1: Often, many rows in Y' are good

Let ε_i be the minimal distance of $\mathsf{Cond}(Y, i)$ from a $\delta' d'$-source. According to Lemma 6.2.1,

$$\mathbb{E}_{i \in [A]}[\varepsilon_i] \leq \varepsilon_{\mathsf{Cond}}.$$

Definition 6.4.4. *We say $z \in \{0,1\}^{d'}$ is* good *if $\mathsf{Raz}(X, z)$ is $\varepsilon_{\mathsf{Raz}}$-close to uniform. Let GZ be the set of all good z-s, and BZ the rest. We say $i \in [A]$ is* good *for $y \in \{0,1\}^d$ if $\mathsf{Cond}(y, i) \in GZ$ and* bad *otherwise. We define a random variable B_i, where the sample space is Y, and $B_i(y) = 1$ if i is bad for y and 0 otherwise.* ◊

By Claim 6.2.4, $|BZ| \leq 2^{\delta_{\mathsf{Raz}} d'}$. Therefore, in expectation, the number of bad rows for y is small:

Claim 6.4.5. $\mathbb{E}_{y \in Y}\left[\sum_{i \in [A]} B_i(y)\right] \leq 2\varepsilon_{\mathsf{Cond}} A$.

Proof. Fix an $i \in [A]$. We have that $\mathsf{Cond}(Y, i)$ is ε_i-close to some $\delta' d' = 0.69d'$-source R. Hence:

$$\mathbb{E}_y[B_i(y)] = \Pr_{y \in Y}[C(y, i) \in BZ] \leq \varepsilon_i + \Pr_{r \in R}[r \in BZ] \leq \varepsilon_i + \frac{|BZ|}{2^{\delta' d'}} = \varepsilon_i + 2^{-0.09d'}.$$

Thus, for d large enough,

$$\mathbb{E}_y\Big[\sum_{i\in[A]} B_i(y)\Big] = \sum_{i\in[A]} \mathbb{E}_y[B_i(y)] \le \sum_{i\in[A]} \Big(\varepsilon_i + 2^{-0.09d'}\Big) \le \varepsilon_{\mathsf{Cond}} A + 2^{-0.09d'} A \le 2\varepsilon_{\mathsf{Cond}} A.$$

□

Definition 6.4.6. *We say* $y \in \mathrm{Supp}(Y)$ has many bad rows *if* $\sum_{i\in[A]} B_i(y) \ge \sqrt{\varepsilon_{\mathsf{Cond}}} A$. ◊

Denote $p_{1,1} = \frac{\varepsilon_0}{4}$.

Claim 6.4.7. $\Pr_{y\in Y}[y \text{ has many bad rows}] \le p_{1,1}$.

Proof. By Markov,

$$\Pr_{y\in Y}\Big[\sum_i B_i(y) \ge \sqrt{\varepsilon_{\mathsf{Cond}}} A\Big] \le \frac{\mathbb{E}\big[\sum_i B_i(y)\big]}{\sqrt{\varepsilon_{\mathsf{Cond}}} A} \le \frac{2\varepsilon_{\mathsf{Cond}} A}{\sqrt{\varepsilon_{\mathsf{Cond}}} A} = 2\sqrt{\varepsilon_{\mathsf{Cond}}} \le \frac{\varepsilon_0}{4},$$

where the last inequality follows from the fact that $\varepsilon_{\mathsf{Cond}} \le (\frac{\varepsilon_0}{8})^2$. □

Part 2: The good rows are t-wise independent

We introduce some notations to simplify the expressions in the proof. For $y_0 \in \{0,1\}^d$ and $k \in [A]$, let $Y'''_k(y_0)$ denote $(Y'''_k | Y = y_0)$. Also, for a set $S \subseteq [A]$, define $Y'''_S(y_0) = \big\{Y'''_j(y_0)\big\}_{j\in S}$. Denote $p_2 = \varepsilon_{\mathsf{Raz}} + 2\varepsilon_{\mathsf{CBA}}$.

Definition 6.4.8. *Let* $y_0 \in \{0,1\}^d$ *(not necessarily in the support of* Y*). Let* $i \in [A]$ *and* $S \subseteq [A] \setminus \{i\}$ *of cardinality* $t-1$. *We say* y_0 violates the product rule *for* (i,S) *if* $B_i(y_0) = 0$ *and*

$$(Y'''_i(y_0), Y'''_S(y_0)) \not\approx_{p_2} U_1 \times Y'''_S(y_0).$$

◊

Definition 6.4.9. *Let* $y_0 \in \{0,1\}^d$ *(not necessarily in the support of* Y*). Let* $i \in [A]$ *and* $S \subseteq [A] \setminus \{i\}$ *of cardinality* $t-1$. *We say* y_0 violates the product rule with distinguisher $\Delta\colon \{0,1\}^t \to \{0,1\}$ *for* (i,S) *if* $B_i(y_0) = 0$ *and*

$$\Big|\Pr[\Delta\left(Y'''_i(y_0), Y'''_S(y_0)\right) = 1] - \Pr[\Delta(U_1, Y'''_S(y_0)) = 1]\Big| > p_2.$$

◊

Observe that if y_0 violates the product rule then there exists some Δ such that y_0 violates the product rule with distinguisher Δ.

Lemma 6.4.10. *For every* i *and* S *as above, the number of* $y \in \{0,1\}^d$ *that violate the product rule for* (i,S) *is at most* $2^{\delta_0 d/2 + 2^t}$.[2]

[2]We could have used an alternative argument that avoids the 2^{2^t} factor here by a minor deterioration in the error of the CBA. However, since the t we use is constant the 2^{2^t} factor is negligible.

Proof. Suppose the lemma is false for some (i, S). Then, by the pigeonhole principle there exists some Δ such that the number of elements $y \in \{0,1\}^d$ that violate the product rule for (i, S) with distinguisher Δ is at least $2^{\delta_0 d/2}$. Let BY denote the set of these elements. Identify BY with the uniform distribution over the set BY.

Let $BY'_i = \mathsf{Cond}(BY, i)$, $BY''_i = \mathsf{Raz}(X, BY'_i)$ and $BY'''_i = \mathsf{CBA}(BY, BY''_i, i)$. For a subset $T \subseteq [A]$ Let BY'_T denote the sub-table of BY' corresponding to the rows of T, and similarly BY''_T and BY'''_T. Since for every $y \in BY$, we have that

$$\Delta\left(BY'''_i(y), BY'''_S(y)\right) \not\approx_{p_2} \Delta(U_1, BY'''_S(y)),$$

this holds also on average, that is

$$\Delta\left(BY'''_i, BY'''_S\right) \not\approx_{p_2} \Delta(U_1, BY'''_S).$$

Thus, it follows that

$$BY'''_{S\cup\{i\}} \not\approx_{p_2} U_1 \times BY'''_S. \tag{6.11}$$

On the other hand, when we condition on the values of

$$\mathcal{H} \;=\; BY'_{S\cup\{i\}}$$

the conditions for the correlation-breaker with advice hold:

- BY and $BY''_{S\cup\{i\}}$ are independent given $\mathcal{H} = BY'_{S\cup\{i\}}$, since $\mathcal{H}$ is a function of BY alone, and given that $\mathcal{H} = BY'_{S\cup\{i\}} = h$ for some h, $BY''_{S\cup\{i\}}$ is a function of X alone.
- $$\begin{aligned}\tilde{H}_\infty(BY|\mathcal{H}) &\geq H_\infty(BY) - \log(|\operatorname{Supp}(\mathcal{H})|)\\ &= H_\infty(BY) - td' \geq \frac{\delta_0 d}{2} - td' \geq \frac{\delta_0 d}{4},\end{aligned}$$

  because

  $$\frac{td'}{d} = t \cdot \left(\frac{2}{3}\right)^s = \frac{\delta_0}{4}.$$

  Now, since $k_{\mathsf{CBA}} = \frac{\delta_0 d}{8}$ and $\varepsilon_{\mathsf{CBA}} = \frac{1}{d}$ we also have for d large enough,

  $$\tilde{H}_\infty(BY|\mathcal{H}) \geq \frac{\delta_0 d}{4} \geq k_{\mathsf{CBA}} + \log\frac{1}{\varepsilon_{\mathsf{CBA}}}.$$

- $B_i(y) = 0$, hence $BY'_i \in GZ$ and $BY''_i = \mathsf{Raz}(X, BY'_i)$ is $\varepsilon_{\mathsf{Raz}}$-close to uniform.

Thus, by the correlation-breaker with advice property,

$$\left(\mathsf{CBA}(BY, BY''_i, i), \left\{\mathsf{CBA}(BY, BY''_j, j)\right\}_{j\in S}\right) \approx_{\varepsilon_{\mathsf{Raz}}+2\varepsilon_{\mathsf{CBA}}} \left(U_1, \left\{\mathsf{CBA}(BY, BY''_j, j)\right\}_{j\in S}\right),$$

or, equivalently,

$$(BY'''_i, BY'''_S) \approx_{p_2} U_1 \times BY'''_S,$$

in contradiction to Equation (6.11). □

Definition 6.4.12. *Say $y \in \{0,1\}^d$* violates the product rule *if it violates it for some $i \in [A]$ and $S \subseteq [A] \setminus \{i\}$ of cardinality $t-1$.* ◊

As $H_\infty(Y) \geq \delta_0 d$, the probability $y \in Y$ violates the product rule for a specific (i,S) is at most $2^{\delta_0 d/2 + 2^t - \delta_0 d} = 2^{2^t - \delta_0 d/2}$. Let $p_{1,2} = \frac{\varepsilon_0}{10}$. Then, by the union bound, for d large enough:

Corollary 6.4.13. $\Pr_{y \in Y}[y \text{ violates the product rule}] \leq 2^{2^t - \delta_0 d/2} \cdot A^t \leq p_{1,2}$.

Part 3: Completing the proof

Definition 6.4.14. *We say y is* bad *if it has many bad rows or if it violates the product rule. If y is not bad we say it is* good. ◊

Let $p_1 = p_{1,1} + p_{1,2}$. Clearly, by Claim 6.4.7 and Corollary 6.4.13, $\Pr_{y \in Y}[y \text{ is bad}] \leq p_1 = \left(\frac{1}{4} + \frac{1}{10}\right)\varepsilon_0$.

Claim 6.4.15. *Fix any good $y \in Y$. Then, $Y'''(y)$ is a (q,t,tp_2) non-oblivious bit-fixing source, for $q = \sqrt{\varepsilon_{\mathsf{Cond}}}A$.*

Proof. Let $Q(y) \subseteq [A]$ be the set of bad rows for y. As y does not have many bad rows, $|Q(y)| = \sum_{i \in [A]} B_i(y) \leq \sqrt{\varepsilon_{\mathsf{Cond}}}A = q$.

Now, fix any set $S \subseteq [A] \setminus Q(y)$ of cardinality t. Let $i \in S$. As $S \subseteq [A] \setminus Q(y)$ and $i \in S$ we have $i \notin Q(y)$ and therefore $B_i(y) = 0$. Also, y does not violate the product rule, hence,

$$\left(Y'''_i(y), Y'''_{S \setminus \{i\}}(y)\right) \approx_{p_2} U_1 \times Y'''_{S \setminus \{i\}}(y).$$

As this is true for any $i \in S$, by Lemma 2.1.2,

$$Y'''_S(y) \approx_{tp_2} U_t.$$

Thus, $Y'''(y)$ is a (q,t,tp_2) non-oblivious bit-fixing source. □

In particular, By Lemma 2.1.11, for every good y, $Y'''(y)$ is $tA^t p_2$-close to a (q,t) non-oblivious bit-fixing source. By the choices we have made above $q = \sqrt{\varepsilon_{\mathsf{Cond}}}A \leq A^{1-2\gamma}$. Equation (6.2) implies that

$$t = \frac{\delta_0}{4}\left(\frac{1}{\varepsilon_{\mathsf{Cond}}}\right)^{\log(3/2)} \geq c_\gamma \log^4 A.$$

Using the resiliency of f from Theorem 2.1.15 (and the fact that it is almost balanced), the output when y is good is p_3-close to uniform for $p_3 = tA^t p_2 + \varepsilon_f + A^{-1/c_\gamma}$, where the first term is due to the distance from a t-wise distribution, the second is due to the resiliency and the third is due to the bias of f (see, e.g., Lemma 2.11 in [CZ16]). To that we also have to add the probability p_1 that y is not good. To finish the proof we notice that:

- It holds that

$$\varepsilon_f \leq c_\gamma \frac{q}{A^{1-\gamma}} \leq c_\gamma \frac{A^{1-2\gamma}}{A^{1-\gamma}} = c_\gamma A^{-\gamma} \leq c_\gamma 2^{-\gamma \log \frac{1}{\varepsilon_{\mathsf{Cond}}}} \leq \frac{\varepsilon_0}{5},$$

  because $\varepsilon_{\mathsf{Cond}} \leq (\frac{\varepsilon_0}{5c_\gamma})^{1/\gamma}$.

	Required entropy k	Seed length d	
Lower-bound, non-explicit	$\log\log(\frac{1}{\varepsilon})$	$\log(\frac{1}{\varepsilon})+\log n$	[RTS00, MRZ14]
[GI02]	$\log\log(\frac{1}{\varepsilon})$	$(2+\gamma)\log(\frac{1}{\varepsilon})+\log n$	Constant ε, or prob. construction
This work (Theorem 6.5.2)	$O(\log\log\frac{1}{\varepsilon})$	$(1+\gamma)\log(\frac{1}{\varepsilon})$	$\mathsf{poly}(1/n)$ error

Table 6.1: Parameters of strong (k,ε) one-bit dispersers, up to additive $O(1)$ terms. γ is an arbitrarily small positive constant.

- Also,
$$A^{-1/c_\gamma} \leq 2^{-\frac{1}{c_\gamma}\log\frac{1}{\varepsilon_{\mathsf{Cond}}}} = {\varepsilon_{\mathsf{Cond}}}^{1/c_\gamma} \leq \frac{\varepsilon_0}{5},$$
because $\varepsilon_{\mathsf{Cond}} \leq (\frac{\varepsilon_0}{5})^{c_\gamma}$.
- Finally, $tp_2 = t(\varepsilon_{\mathsf{Raz}}+2\varepsilon_{\mathsf{CBA}})$, $\varepsilon_{\mathsf{Raz}} = 2^{-\Omega(k)} = d^{-\Omega(1)}$, $\varepsilon_{\mathsf{CBA}} = \frac{1}{d}$. Thus, $tp_2 \leq 4td^{-\Omega(1)}$. A and t are constants, so for d large enough, $tA^tp_2 \leq \frac{\varepsilon_0}{5}$.

Together, the error is at most $p_1+p_3 \leq \varepsilon_0$ completing the proof of the theorem. □

6.5 Strong Seeded Dispersers and Friends

6.5.1 Strong seeded dispersers

Definition 6.5.1. $\mathsf{Disp}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a strong* (k,ε) *disperser, if for every* (n,k) *source* X*,*
$$|\operatorname{Supp}((Y,\mathsf{Disp}(X,Y)))| > (1-\varepsilon)DM.$$
We say Disp *is (source) linear if for every* $y \in \{0,1\}^d$ *and every* $x_1,x_2 \in \mathbb{F}_2^n$, $\mathsf{Disp}(x_1+x_2,y) = \mathsf{Disp}(x_1,y)+\mathsf{Disp}(x_2,y)$. ◊

We are interested in the important special case where $m=1$. In this case, non-explicitly, a random function is (w.h.p.) a strong (k,ε) disperser with $d = \log n + \log(\frac{1}{\varepsilon}) + O(1)$ provided that $k \geq \log\log(\frac{1}{\varepsilon}) + O(1)$ [RTS00, MRZ14]. A matching lower bound, up to additive constant factors, was given by [RTS00].

Using the translation between strong seeded dispersers and erasure list-decodable codes which we discuss in Section 6.5.3, Guruswami and Indyk's result [GI02] gives a probabilistic polynomial time algorithm that outputs with high probability a strong seeded disperser with seed-length $d = 2\log(\frac{1}{\varepsilon})+\log n+\log\log(\frac{1}{\varepsilon})$ and optimal entropy loss. The construction can be made deterministic, but with running time exponential in $1/\varepsilon$. See Table 6.1 for a summary of previous results.

Note that as we discuss the one output bit case, the required entropy is essentially the *entropy loss*. Form Theorem 6.4.1 we can derive a better explicit construction of a strong disperser with low error.

Theorem 6.5.2. *For every constant $0 < \gamma < 1$ there exists a constant $c \geq 1$ such that for every integer n and every $\varepsilon \leq n^{-\frac{c}{1-\gamma}}$ there exists an explicit strong (k, ε) disperser $\mathsf{Disp}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$ where $d = (1+\gamma)\log(\frac{1}{\varepsilon})$ and $k = c \log d$.*

Proof. Set $\varepsilon_0 = \frac{1}{4}$ and $\delta_0 = \frac{\gamma}{1+\gamma}$. Let c be the constant from Theorem 6.4.1 for δ_0 and ε_0 and let $\mathsf{2Ext}\colon [N] \times [D] \to \{0,1\}$ be the $((n,k),(d,k_2 = \delta_0 d),\varepsilon_0)$ two-source extractor where $d = (1+\gamma)\log(\frac{1}{\varepsilon})$ and $k = c \log d$. Notice that $d \geq c \log n$ (because $\varepsilon \leq n^{-\frac{c}{1+\gamma}}$) as required. Let $\mathsf{Disp}(x,y) = \mathsf{2Ext}(x,y)$.

Let $X \subseteq [N]$ be a set of size K and call a value $y \in [D]$ *b-bad* if $\mathsf{Disp}(X,y) = \{b\}$. It follows that the sets of 0-bad y-s and 1-bad y-s are each of size less than K_2. Therefore,

$$|\operatorname{Supp}\left((U_d, \mathsf{Disp}(X,U_d))\right)| > 2K_2 + 2(D - 2K_2) = 2D - 2K_2 = \left(1 - \frac{K_2}{D}\right) 2D = (1-\varepsilon)2D,$$

because $\frac{K_2}{D} = 2^{-(1-\delta_0)d} = 2^{-\frac{1}{1+\gamma}d} = 2^{-\log(\frac{1}{\varepsilon})} = \varepsilon$. □

6.5.2 Non-strong dispersers

We now prove Theorem 6.1.5 and output more bits from the source at the expense of being strong in only most of the seed. We construct

$$\mathsf{Disp}\colon \{0,1\}^n \times \{0,1\}^{d_1} \times \{0,1\}^{d_2} \to \{0,1\}^m,$$

where we think of d_1 and d_2 as two parts of the seed. Disp will be strong in the first d_1 bits of the seed. Using the notations of Section 6.4 we let

$$\mathsf{Disp}(x,y,i) = (y, \mathsf{Raz}(x, \mathsf{Cond}(y,i))).$$

We now prove (in sketch) Theorem 6.1.5.

Proof. We adopt the notations of Section 6.4. In those notations, $\mathsf{Disp}(X,Y,I) = (Y, Y''_I)$. First note that the length of $i \in \{0,1\}^{d_2}$ is the logarithm of the number of rows in the table Y'' which is $a = O(1)$. By Claim 6.4.7 we know that for nearly every $y \in \{0,1\}^{d_1}$ we have many values $i \in \{0,1\}^{d_2}$ such that $\mathsf{Raz}(X, \mathsf{Cond}(y,i))$ is $\varepsilon_{\mathsf{Raz}}$-close to uniform. In particular, for every y that has many good rows, let i_y be any such row. Then,

$$\begin{aligned} |\operatorname{Supp}(\mathsf{Disp}(X, U_{d_1}, U_{d_2}))| &\geq \sum_{y \text{ has many good rows}} |\operatorname{Supp}(\mathsf{Disp}(X,y,i_y))| \\ &\geq \sum_{y \text{ has many good rows}} (1 - \varepsilon_{\mathsf{Raz}}) 2^{d''}. \end{aligned}$$

The theorem now follows since $d'' = \Omega(k)$ and $\varepsilon_{\mathsf{Raz}}$ is smaller than $2^{-\Omega(d'')}$, which implies that we can truncate the output of Raz such that when $\mathsf{Raz}(X, \mathsf{Cond}(y,i))$ is $\varepsilon_{\mathsf{Raz}}$-close to uniform it covers its entire support. □

	Rate $R = n/\bar{n}$	List size L	
Lower-bound, non-explicit	ε	$\log(\frac{1}{\varepsilon})$	[Gur03]
[GI02]	$\frac{\varepsilon^2}{\log(1/\varepsilon)}$	$\log(\frac{1}{\varepsilon})$	Constant ε, or prob. construction
This work (Theorem 6.5.8)	$\varepsilon^{1+\gamma}$	$\log^{O(1)}(\frac{1}{\varepsilon})$	$\mathsf{poly}(1/n)$ error

Table 6.2: Parameters of $(\bar{n}, N)_2$ codes, $((1-\varepsilon)\bar{n}, L)$ erasure list-decodable, up to constant multiplicative factors. γ is an arbitrarily small positive constant.

6.5.3 Erasure list-decodable codes

An $(\bar{n}, n)$ (binary) code is a mapping $\mathcal{C}\colon \{0,1\}^n \to \{0,1\}^{\bar{n}}$. The code $\mathcal{C}$ is *linear* if $\mathcal{C}$ is linear, and is denoted by $[\bar{n}, n]$. We identify a code with the image of $\mathcal{C}$. For a linear $\mathcal{C}$ this image is a linear subspace of $\mathbb{F}_2^{\bar{n}}$ of dimension n. A generator matrix for an $[\bar{n}, n]$ code $\mathcal{C}$ is any matrix whose columns form a basis for $\mathcal{C}$. In the *erasures* noise model, an adversarially chosen subset of the codeword's symbols are erased and the positions where erasures have occurred are known.

Definition 6.5.3. *A code $\mathcal{C} \subseteq \{0,1\}^{\bar{n}}$ is (s, L) erasure list-decodable if for every $r \in \{0,1\}^{\bar{n}-s}$ and every set $T \subseteq [\bar{n}]$ of size $\bar{n} - s$,*

$$\left|\left\{c \in \mathcal{C} \;\middle|\; c|_T = r\right\}\right| \;<\; L,$$

where $c|_T$ denotes the projection of c to the coordinates in T. ◊

The following folklore lemma (see, e.g., [Gur03, Lemma 1]) gives an alternative characterization of *linear* erasure list-decodable codes.

Lemma 6.5.4. *An $[\bar{n}, n]_2$ linear code $\mathcal{C}$ is $((1-\varepsilon)\bar{n}, L)$ erasure list-decodable if and only if its $\bar{n} \times n$ generator matrix G has the property that every $\varepsilon\bar{n} \times n$ sub-matrix of G has rank greater than $n - \log L$.*

Non-explicitly, we have:

Theorem 6.5.5 ([Gur03])**.** *For every n and $\varepsilon > 0$, there exists an $(\bar{n}, n)$ binary code that is $((1-\varepsilon)\bar{n}, L)$-erasure list-decodable of rate $\frac{n}{\bar{n}} = \Omega(\varepsilon)$ and $L = O(\log \frac{1}{\varepsilon})$.*

See Table 6.2 for a summary of previous results.

Guruswami [Gur04a] observed that strong dispersers can be used to construct erasure list-decodable codes. Here we complement his argument, and note that strong dispersers are *equivalent* to erasure list-decodable codes. Given a function $\mathsf{Disp}\colon [N] \times [D] \to \{0,1\}$, we consider the (D, n) code $\mathcal{C}_{\mathsf{Disp}}\colon \{0,1\}^n \to \{0,1\}^D$ defined by $\mathcal{C}_{\mathsf{Disp}}(x)_i = \mathsf{Disp}(x, i)$. Note that the code is linear if and only if Disp is linear.

Lemma 6.5.6 (following [Gur04a])**.** *The function $\mathsf{Disp}\colon [N] \times [D] \to \{0,1\}$ is a strong (k, ε) disperser if and only if $\mathcal{C}_{\mathsf{Disp}}$ is $((1-2\varepsilon)D, K)$ erasure list-decodable.*

Proof. For one direction, assume Disp is a strong (k, ε) disperser. We wish to prove that $\mathcal{C}_{\mathsf{Disp}}$ is $((1-2\varepsilon)D, K)$ erasure list-decodable. Let $T = \{t_1, \ldots, t_{2\varepsilon D}\} \subseteq [D]$ be an arbitrary set of size $2\varepsilon D$ and $r \in \{0,1\}^{2\varepsilon D}$ an arbitrary string. Let $X_{T,r} \subseteq \{0,1\}^n$ denote the set of all the messages x for which $\mathcal{C}_{\mathsf{Disp}}(x)|_T = r$. Then,

$$|\operatorname{Supp}\left((U_d, \mathsf{Disp}(X_{T,r}, U_d))\right)| \leq |T| \cdot 1 + (D - |T|) \cdot 2 \leq (1-\varepsilon)2D,$$

where the first inequality follows by considering seeds in T and seeds in $[D] \setminus T$. For a seed $t_i \in T$ we have that $\mathsf{Disp}(X_{T,r}, t_i)$ is fixed, hence each such seed contributes 1 to the support size. For any other seed y, the support size of $\mathsf{Disp}(X_{T,r}, y)$ is at most 2. As Disp is a strong (k, ε) disperser, we conclude that $|X_{T,r}| \leq K$ as desired.

For the other direction assume Disp is a not a strong (k, ε) disperser. Then, there exists a set $X \subseteq \{0,1\}^n$ such that $|X| \geq K$ and $|\operatorname{Supp}\left((U_d, \mathsf{Disp}(X, U_d))\right)| \leq (1-2\varepsilon)2D$. Note that for every $y \in [D]$ we have $|\operatorname{Supp}(\mathsf{Disp}(X, y))| \in \{1, 2\}$. Therefore, following the above calculation, there exists a set $T \subseteq D$ of size at least $2\varepsilon D$ such that for each $y \in T$, $|\operatorname{Supp}(\mathsf{Disp}(X, y))| = 1$. But this means that for every $x \in X$, $\mathcal{C}_{\mathsf{Disp}}(x)|_T$ is the same (punctured) codeword. It follows that $\mathcal{C}_{\mathsf{Disp}}$ is not $((1-2\varepsilon)D, K)$ erasure list-decodable. □

Corollary 6.5.7. *If* $\mathsf{Disp}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$ *is a strong* (k, ε) *disperser with seed-length* $d = a_1 \log n + a_2 \log(\frac{1}{\varepsilon}) + a_3$ *(for some* $a_1 \geq 1, a_2 \geq 1$ *and* a_3*) then* $\mathcal{C}_{\mathsf{Disp}}$ *is a* $((1-2\varepsilon)D, K)$ *erasure list-decodable code of rate* $2^{-a_3} \cdot n^{1-a_1} \cdot \varepsilon^{a_2}$.

When ε is much smaller than $\frac{1}{n}$ the dominant factor is determined by a_2. As we mentioned earlier (and as Guruswami also notes in [Gur04a]) previous explicit constructions for binary codes had $a_2 \geq 2$ (usually inherited from extractor constructions). Our construction is the first to get arbitrary close to $a_2 = 1$ and small list-size. Combining Corollary 6.5.7 and Theorem 6.5.2, we obtain:

Theorem 6.5.8. *For every constant* $0 < \gamma < 1$ *there exists a constant* $c \geq 1$ *such that for every integer* n *and every* $\varepsilon \leq n^{-\frac{c}{1-\gamma}}$ *there exists an explicit code* $\mathcal{C}\colon \{0,1\}^n \to \{0,1\}^{(\frac{1}{\varepsilon})^{1+\gamma}}$ *that is*

$$\left((1-2\varepsilon)\left(\frac{1}{\varepsilon}\right)^{1+\gamma}\ ,\ \left((1+\gamma)\log\frac{1}{\varepsilon}\right)^c\right)$$

erasure list-decodable of rate $n\varepsilon^{1+\gamma}$.

6.5.4 Ramsey graphs

In this section we tackle the problem of constructing *unbalanced* Ramsey graphs.

Definition 6.5.9. *A bipartite graph* $\mathsf{Ram}\colon [N_1] \times [N_2] \to \{0,1\}$ *is a* (K_1, K_2) *bipartite Ramsey graph if every* $K_1 \times K_2$ *induced subgraph of* Ram *is neither a bipartite clique nor a bipartite independent set.* ◊

While it is possible to interpret some pseudorandom objects as unbalanced Ramsey graphs, they were less studied explicitly. See Table 6.3 for a summary of previous results.

It is easy to see that a two-source extractor with any non-trivial error is, in fact, a bipartite Ramsey graph, so as a corollary of Theorem 6.4.1, we obtain:

	$K_1 : N_1$	$K_2 : N_2$	
Lower-bound	$(c-1)\log n : 2^n$	$n : n^c$	[RTS00] and Claim 6.5.11
Non-explicit	$O(c \log n) : 2^n$	$n : n^c$	Probabilistic method
[Raz05]	$\log^{O(1)} n : 2^n$	$N_2^{0.5+\gamma} : n^{O(1)}$	$O(1)$ terms depend on γ
This work (Theorem 6.4.1)	$\log^{O(1)} n : 2^n$	$N_2^{\gamma} : n^{O(1)}$	$O(1)$ terms depend on γ

Table 6.3: Parameters of (K_1, K_2) Ramsey graphs in the unbalanced case, $[N_1 = 2^n] \times [N_2]$. c is any large enough constant and γ is an arbitrarily small positive constant.

Corollary 6.5.10. *For every integer N_1 and a constant $0 < \delta < 1$ there exists a constant $c = c(\delta) \geq 1$ and an explicit function $\mathsf{Ram} : [N_1] \times [N_2] \to \{0,1\}$ that is a bipartite $(K_1, K_2 = N_2^{\delta})$ Ramsey graph, for $N_2 = \log^c N_1$ and $K_1 = \log^c N_2$.*

We start with the easy claim that bipartite Ramsey graphs are equivalent to strong one-bit dispersers.

Claim 6.5.11. *If $\mathsf{Ram} : [N_1] \times [N_2] \to \{0,1\}$ is a (K_1, K_2) bipartite Ramsey graph then Ram is a strong $(k_1, \varepsilon \geq \frac{K_2}{N_2})$ disperser with seed-length $n_2 = k_2 + \log(\frac{1}{\varepsilon})$. Also, if Ram is a strong $(k_1, \varepsilon = \frac{K_2}{2N_2})$ disperser then it is a (K_1, K_2) bipartite Ramsey graph.*

Proof. The first claim follows from the proof of Theorem 6.5.2.

For the other claim, which was already observed in [GKRTS05], assume Ram is a $(k_1, \varepsilon = \frac{K_2}{2N_2})$ disperser and assume towards contradiction that it is not a $(K_1, K_2 = 2\varepsilon N_2)$ bipartite Ramsey graph. Hence, there exist some $S \subseteq [N_1]$ and $T \subseteq [N_2]$ so that $|S| \geq K_1$ and $|T| \geq K_2$ such that either $\mathsf{Ram}(S,T) = \{0\}$ or $\mathsf{Ram}(S,T) = \{1\}$. Assume without loss of generality that $\mathsf{Ram}(S,T) = \{0\}$, so for every $t \in T$, $(t,1) \notin \mathrm{Supp}\left((U_{n_2}, \mathsf{Ram}(S, U_{n_2}))\right)$. But then,

$$|\mathrm{Supp}\left((U_{n_2}, \mathsf{Ram}(S, U_{n_2}))\right)| \leq 2(N_2 - |T|) + |T| \leq (1-\varepsilon)2N_2,$$

a contradiction. □

As observed in [GKRTS05], the quality of the Ramsey graph implied by the above theorem crucially depends on the seed-length of the given disperser. Specifically, if the seed-length dependence on the error ε is $2 \cdot \log(\frac{1}{\varepsilon})$ then $K_2 = 2\varepsilon N_2 > \sqrt{N_2}$ and if it is $1 \cdot \log(\frac{1}{\varepsilon})$ then K_2 can be very small.

We mention a more frugal way of obtaining Ramsey graphs from *linear* dispersers. The argument is a straightforward adaptation of an argument of Alon [Gur04b, Proposition 10.15].[3] The parameters we obtain are identical to the above claim (and [GKRTS05]), except that one side of the graph is scaled down (from N to n) as is its entropy (from K to k).

Theorem 6.5.12. *Suppose $\mathsf{Disp} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$ is a linear strong (K, ε) disperser. Let G be the $D \times n$ generating matrix of the $[D, n]_2$ linear code $\mathcal{C}_{\mathsf{Disp}}$. Then, G is a $(2\varepsilon D, k+1)$ bipartite-Ramsey-graph.*

[3] Alon's argument is aimed at obtaining *balanced* Ramsey graphs, while we are more concerned with the entropy they can handle.

Proof. Assume Disp is a linear strong (K, ε) disperser. By Lemma 6.5.6, $\mathcal{C}_{\mathsf{Disp}}$ is a $((1-2\varepsilon)D, K)$ erasure list-decodable code. Assume towards contradiction that G is not a $(2\varepsilon D, k+1)$ bipartite Ramsey graph. Let M' be a monochromatic $2\varepsilon D \times k+1$ sub-matrix of G. Assume that M' is the all-ones matrix (a similar argument handles the all-zeros matrix). Denote by M the $2\varepsilon D \times n$ sub-matrix of G that is formed by taking the rows of M' and all columns of G. On the one hand, by Lemma 6.5.6 and Lemma 6.5.4, $\text{rank}(M) > n - \log K = n - k$. On the other hand, as M contains $k+1$ columns of rank 1, $\text{rank}(M) \leq n - k$, a contradiction. □

It is natural to ask whether the other direction also holds, namely whether an adjacency matrix of a bipartite Ramsey graph is in fact a generating matrix of a linear, erasure list-decodable code. Stated differently, whether a low-rank matrix must contain large monochromatic rectangles. That question received much attention, as it is tightly related to the famous "log-rank conjecture" in communication complexity [Lov14, NW95]. Unfortunately, the acclaimed unconditional upper bound of Lovett [Lov16] still does not give us a meaningful result.

6.6 Concluding Remarks and Open Problems

- The strong disperser we construct in this chapter outputs one bit, and for min-entropy $k = O(\log\log\frac{1}{\varepsilon})$,
  - has $O(\log\log\frac{1}{\varepsilon})$ entropy loss, and,
  - $(1+\gamma)\cdot\log(\frac{1}{\varepsilon})$ dependence of the seed-length on the error.

  It is natural to ask to extend the results of the chapter to arbitrarily large values of k, matching (up to multiplicative factors) the non-explicit results.

- Our dispersers are inherently non-linear, and therefore we also get non-linear erasure list-decodable codes. How can we obtain near optimal *linear* codes?

- The erasure list-decodable code we construct is explicit in the sense that the code can be efficiently encoded. Does it also admit an efficient erasure list-*decoding* algorithm?

- The seed-length of our strong disperser is $c\log n + \log(\frac{1}{\varepsilon})$. Pushing c closer to 1 is an important open problem. In particular it would imply erasure list-decodable codes of near-optimal rate even for relatively large ε. Such a disperser with many output bits can also be used for simulating one-sided error randomized algorithms using weak random sources with nearly linear overhead [Zuc96b].

Part II

Probabilistic Small-Space Computation

Chapter 7

Part II Overview

One of the most fundamental questions in complexity theory is whether one can save resources like space and time by using randomness. In this line of research, an important problem is derandomizing space-bounded probabilistic computations. Space-bounded probabilistic complexity classes are the focal point of this part of the thesis, so let us define the computational model rigorously.

Space-bounded computation. A deterministic space-bounded Turing machine has three semi infinite tapes: an *input tape* (that is read-only), a *work tape* (that is read-write) and an *output tape* (that is write-only and uni-directional). The space complexity of the machine is the number of cells on the work tape. The running time of a space-bounded Turing machine with $s(n) \geq \log n$ space complexity is bounded by $2^{O(s(n))}$ time. A *probabilistic* space-bounded Turing machine is similar to the deterministic machine (and in particular we require it always halt within $2^{O(s(n))}$ time) except that it can also toss random coins. One convenient way to formulate this is by adding a fourth semi-infinite tape, the *random-coins tape*, that is read-only, uni-directional and is initialized with perfectly uniform bits. We are only concerned with bounded-error computation: We say a language is accepted by a probabilistic Turing machine if for every input in the language the acceptance probability is at least 2/3, and for every input not in the language it is at most 1/3. As usual, the acceptance probability can be amplified as long as there is some non-negligible gap between the acceptance probability of yes and no instances.

Definition 7.0.1. *A language is in* $\mathsf{BPSPACE}(s(n))$ *if it is accepted by a probabilistic space bounded TM with space complexity* $s(n)$. $\mathsf{BPL} = \cup_c \mathsf{BPSPACE}(c \log n)$. $\Diamond$

We say a language is accepted by a probabilistic Turing machine with one-sided error if for every input in the language the acceptance probability is at least 1/2, and for every input not in the language we always reject.

Definition 7.0.2. *A language is in* $\mathsf{RSPACE}(s(n))$ *if it is accepted by a probabilistic space bounded TM with one-sided error and space complexity* $s(n)$. $\mathsf{RL} = \cup_c \mathsf{RSPACE}(c \log n)$. $\Diamond$

To study space-bounded probabilistic computation it is enough to concentrate on the small-space classes, and there we know that

$$\mathsf{NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{RL} \subseteq \mathsf{BPL} \subseteq \mathsf{NC}^2 \subseteq \mathsf{L}^2.$$

Under hardness assumptions[1] it holds that $\mathsf{L} = \mathsf{RL} = \mathsf{BPL}$, and a full derandomization of probabilistic logspace has been a major challenge in complexity theory for many years.

The *pseudorandomness* approach to this problem has lead to many unconditional results. The first pseudorandom generator[2] (PRG) for BPL was given in the seminal work of Nisan [Nis92, Nis94], eventually yielding $\mathsf{BPL} \subseteq \mathsf{DTISP}(\mathsf{poly}(n), O(\log^2 n))$ (and also a quasi-polynomial universal traversal sequence for general undirected graphs). Several variants of Nisan's generator followed, most prominently Impagliazzo, Nisan and Wigderson's generator [INW94] and the generator by Nisan and Zuckerman [NZ96]. The INW PRG [INW94], although giving similar parameters to those in [Nis92] for polynomial-width read-once branching programs, used a different analysis using expander graphs, and turned out to be more flexible, giving better parameters for some weaker classes of functions [RV05, De11, KNP11, Ste12, BRRY14]. The Nisan-Zuckerman PRG [NZ96] was the first to use seeded extractors, and the parameters it achieves allow for a randomness-time tradeoffs. One conclusion of their work is that any language solvable by a BPL machine using only poly-logarithmic number of random bits already belong to L.

The main building-block in Nisan's PRG is a generator that fools *matrix squaring* (or, put differently, fools two steps of a bounded-width branching program). Saks and Zhou [SZ99a] gave an improved approximate matrix squaring algorithm and showed that $\mathsf{BPL} \subseteq \mathsf{L}^{3/2}$. Roughly, they overcame the dependence among different recursive levels in Nisan's generator, which allowed the use of fewer hash functions.

No improvement upon the [SZ99a] result has been made since, and much effort has been invested in constructing PRGs for more restricted classes of functions in RL. This includes various restrictions on branching programs [BRRY10, De11, BDVY13, CHHL18, MRT18], read-once formulas, linear and polynomial tests, threshold functions, modular tests, combinatorial rectangles and their generalizations combinatorial shapes and Fourier shapes ([GMR+12, LRTV09, NN93, AGHP92, Vio09, ASWZ96, GMRZ13, GKM15, DGJ+10, GOWZ10, MZ13] constitutes a very partial list). Typically, these constructions yield an almost optimal seed-length.

A problem complete for BPL. Other than the *pseudorandomness* problem of constructing PRGs, one can naturally think of *derandomizing* canonical problem and in the small-space setting it is very natural to consider st-connectivity in various contexts. A classical result of Savitch [Sav70] shows that directed connectivity can be solved in L^2, thus showing that $\mathsf{NL} \subseteq \mathsf{L}^2$. A major progress was made by Aleliunas et. al. [AKL+79] who gave (the natural) randomized logspace algorithm for *undirected* connectivity.

Building upon the zig-zag product [RVW02], Reingold [Rei08] derandomized the problem of undirected connectivity by cleverly transforming the input graph into a constant degree expander. His technique also implies a universal traversal sequence for a certain family of graphs. Note that although, in some sense, the work of Reingold (and subsequently, also the work of [RV05]) derandomizes graph powering, it only preserves connectivity and does

[1] Klivans and van Melkebeek [KVM02] proved that if there is a language in linear space that requires circuits of size $2^{\Omega(n)}$, then $\mathsf{BPL} = \mathsf{L}$.

[2] Formally, an ε-error PRG for a class of functions $\mathcal{F} \subseteq \{0,1\}^n \to \{0,1\}$ is a function $G\colon \{0,1\}^d \to \{0,1\}^n$ such that for every $f \in \mathcal{F}$, $|\mathbb{E}[f(U_n)] - \mathbb{E}[f(G(U_d))]| \leq \varepsilon$.

not readily imply a logspace approximate matrix powering. The result of Reingold was extended to directed *regular* (or more generally, Eulerian) graphs by Reingold, Trevisan and Vadhan [RTV06], and a new complete problem for RL was found, namely solving directed connectivity in rapidly-mixing graphs.

Unlike NL and RL, in BPL there were no natural complete problems. Moreover, we had no good candidates for problems in BPL which were not already in L. In [DSTS17], we gave a natural problem that is complete for BPL – approximating the spectral gap of stochastic operators with a real second eigenvalue.[3] This is done by converting a BPL machine to some layered directed graph and then analyze its spectrum. Approximating the spectral gap is a well-known and important problem, and the study of linear-algebraic problems in the context of space-bounded computation may give rise to new insights, as we exemplify next.

Linear algebra in small space and solving Laplacian systems. The complexity class DET is the class of languages that are NC^1 Turing-reducible to the problem *intdet* of computing the determinant of an integer matrix (see [Coo85] for the exact details). It is known that $\mathsf{NL} \subseteq \mathsf{DET} \subseteq \mathsf{NC}^2$. As it turns out, many important problems in linear algebra, such as inverting a matrix, or equivalently, solving a set of linear equations are in DET, and often complete for it (see, e.g., [Coo85]). The fact that $\mathsf{NL} \subseteq \mathsf{DET}$ is due to Cook [Coo85] who showed that the directed connectivity problem is reducible to *intdet*. $\mathsf{DET} \subseteq \mathsf{NC}^2$ follows from Csanky's algorithm [Csa76] for the parallel computation of the determinant. In addition to the above, we also know that $\mathsf{BPL} \subseteq \mathsf{DET}$.

The problem of matrix powering is complete for DET. Also, one can show that *approximating* matrix powering of *stochastic* matrices is in BPL. Conversely, by converting a BPL machine to a stochastic operator A such that the probability the machine moves from s to t in k steps is $A^k[s,t]$, we can see that approximating matrix-powering of stochastic operators is complete for BPL.

Towards understanding probabilistic space-bounded computation, one can think of identifying the supposed *strength* of space-bounded computation, and linear-algebraic problems are promising candidates. A few years ago, Ta-Shma [TS13] showed that it is possible to approximate the SVD of a given matrix (and consequently its inverse) to within polynomially-small accuracy in BQL, the quantum analogue of BPL. In [DTS15b], we addressed the natural question of "de-quantumization": can we approximate the singular values already in BPL? We gave an affirmative answer for the restricted setting of stochastic Hermitian matrices and constant accuracy. Fefferman and Lin [FL18] gave two complete problems for BQL – approximating the inverse and the minimum eigenvalue of positive semi-definite matrices (both to polynomially-small accuracy).

In Chapter 8, we address an important relaxation of matrix inversion – approximating a solution to a Laplacian system. We give a probabilistic logspace algorithm for undirected graphs and rapidly-mixing directed graphs. The algorithm achieves polynomially-small accuracy, similar to the quantum algorithm. In terms of *deterministic* space complexity, following [SZ99a], our work gives an algorithm for approximating a solution to a Laplacian system using $O(\log^{1.5} n)$ space. This was later improved, using different techniques, to

[3]The results of [DSTS17] are not covered in this thesis and are mentioned here to help illustrate the overall approach.

$O(\log n \log\log n)$ by Murtagh et al. [MRSV17].

Our algorithm for approximating a solution to a Laplacian system is in fact an algorithm for approximating a Laplacian's *generalized inverse*. We approximate the generalized inverse by first handling the non-trivial kernel of the operator and then employ techniques from random walks over directed graphs to approximate the inverse of the Laplacian's invertible subspace.

The message emerging from the above discussion is that, somewhat surprisingly, the deterministic, probabilistic and quantum space-bounded classes can be characterized by linear-algebraic promise problems that approximate the exact computation that can be done in DET. The different classes differ in the type of linear operators they can handle. As such, their relationship to the class DET is similar to the relationship between BPP, which can approximate the permanent function [JSV04], and the class #P which solves it exactly [Val79]. We believe this view is not only important by itself (by giving algorithms approximating natural problems in linear algebra), but may also shed new light on the strengths and weaknesses of the deterministic, probabilistic and quantum models of space-bounded computation.

Space-Bounded Approximate-Counting Problems. There are several natural open problems that are still wide open. Is it sill BPL-hard to approximate the spectral gap of stochastic and Hermitian operators (i.e., undirected graphs)? Or is it possible to approximate in BPL the second eigenvalue of a general (not necessarily stochastic or non-negative) operator?

More generally, is it possible to approximate the SVD of an arbitrary linear operator already in BPL? A positive answer would imply BPL approximations to many problems in linear algebra that are currently only known to be in DET. In Chapter 9 we prove that a negative answer would imply a separation between BQL and BPL. More broadly, we discuss the derandomization of approximate-counting problems and show that in the space-bounded setting an inapproximability result readily translates to a separation of the corresponding promise decision classes.

Chapter 8 follows [DLGTS17] and Chapter 9 is due to [DTS15a].

Chapter 8

Probabilistic Logspace Algorithms for Laplacian Solvers

Interested in the study of approximating solutions to linear-algebraic problems in small space, in this chapter we explore the problem of solving Laplacian linear systems. A recent series of breakthroughs initiated by Spielman and Teng culminated in the construction of nearly linear time Laplacian solvers, approximating the solution of a linear system $\mathcal{L}x = b$, where $\mathcal{L}$ is the normalized Laplacian of an undirected graph. Here, we study the *space complexity* of the problem. Surprisingly we are able to show a probabilistic, *logspace* algorithm solving the problem. We further extend the algorithm to other families of graphs like Eulerian graphs (and directed regular graphs) and graphs that mix in polynomial time.

Our approach is to pseudo-invert the Laplacian, by first "peeling-off" the problematic kernel of the operator, and then to approximate the inverse of the remaining part by using a Taylor series. We approximate the Taylor series using a previous work and the special structure of the problem. For directed graphs we exploit in the analysis the Jordan normal form and results from matrix functions.

8.1 Introduction

Approximating the solution of a linear system $\mathcal{L}x = b$, where $\mathcal{L}$ is the normalized *Laplacian* of a graph G, is an important algorithmic challenge with multitude of algorithmic applications (see [Vis13] and references therein). In the time-bounded setting this problem has drawn a lot of attention over the past decade. A series of breakthroughs initiated by Spielman and Teng culminated in the construction of almost linear-time algorithms [KOSZ13, PS14, ST04, ST11, ST13, ST14a].

As said, we are interested in studying the space complexity of this problem, and specifically achieving a probabilistic logspace algorithm that approximates a solution to such a system. We show that the class BPL is powerful enough to approximate the solution to a linear system of equations for a wide and important variety of linear operators, and in particular for Laplacians of undirected graph (which is the focus of the work of Spielman and Teng). In fact we do more and approximate a *generalized inverse* of the Laplacian, i.e., a matrix $\mathcal{L}^\star$ such that $\mathcal{L}\mathcal{L}^\star\mathcal{L} = \mathcal{L}$, which is sufficient for solving such a set of equations. In

essence this means that we invert the matrix on the subspace defined by its image, leaving the kernel unchanged. We prove:

Theorem 8.1.1. *There exists a probabilistic algorithm that gets as input an $n \times n$ stochastic matrix $\mathcal{S}$ that is the transition matrix of an undirected graph and desired accuracy and confidence parameters $\varepsilon, \delta > 0$, and outputs with probability at least $1 - \delta$ an approximation of the generalized inverse $\mathcal{L}^\star = (\mathcal{I} - \mathcal{S})^\star$ to within an ε-accuracy, using*

$$O\left(\log \frac{n}{\varepsilon} + \log\log \frac{1}{\delta}\right)$$

space.

Our goal is to approximate $f(\mathcal{S})$ where f is the function corresponding to the generalized inverse of $\mathcal{I} - \mathcal{S}$. We begin by considering the simpler case where f has a Taylor expansion.

Let G be a regular undirected graph with an associated transition matrix $\mathcal{S}$. As G is undirected and regular, $\mathcal{S}$ is normal and we can represent it as $\mathcal{S} = V\Sigma V^\dagger$ where Σ is a diagonal matrix with the eigenvalues of $\mathcal{S}$ lying on the diagonal. Consider a function f with a Taylor expansion $f(x) = \sum_i c_i x^i$. We would like to approximate $f(\mathcal{S}) = \sum_i c_i \mathcal{S}^i = V f(\Sigma) V^\dagger$.[1] Using Taylor expansion in the space-bounded setting is appealing, as in BPL we can approximate powers of stochastic matrices. Hence, if the series expansion of f behaves "nicely", we can also approximate $f(\mathcal{S})$ in BPL. Using this approach we can, e.g., approximate the matrix $e^{\mathcal{S}}$ using the Taylor expansion $e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$.

We now consider the real problem which is approximating a generalized inverse of the Laplacian $\mathcal{L} = \mathcal{I} - \mathcal{S}$. This means that we want to invert $\mathcal{L} = \mathcal{I} - \mathcal{S}$ on its image, leaving the kernel unchanged. Thus, the function f we want to compute is $\frac{1}{1-x}$ when $x \neq 1$ and 1 otherwise (think of x here as an eigenvalue of $\mathcal{S}$). The function f is not continuous and so does not have a Taylor series around 1. Also notice that the operator $\mathcal{L}$ always has a non-trivial kernel (1 is always an eigenvalue of $\mathcal{S}$). Thus, we cannot directly employ the Taylor series approach.

Our solution to the problem is to first "peel-off" the 1-eigenspace using the stationary distribution of the corresponding random walk on G. We are then left with an invertible operator $\mathcal{I} - \mathcal{A}$ whose eigenvalues are bounded away from 0. We now wish to use the Taylor series approach and approximate $(\mathcal{I} - \mathcal{A})^{-1}$ by $\sum_{i=0}^{\infty} \mathcal{A}^i$, which corresponds to the Taylor series $\frac{1}{1-x} = \sum_{i=0}^{\infty} x^i$. There is yet one obstacle we need to overcome, which is that the operator $\mathcal{A}$ that we get after peeling off the stationary distribution of G, is *not* stochastic, and in fact has ℓ_∞ norm larger than 1. Thus, offhand, we do not necessarily know how to simulate high powers of it in BPL. Nevertheless, we exploit its unique structure and show it can be simulated in BPL. Finally, by recovering the peeled-off layer, we essentially recover the required operator $\mathcal{L}^\star$.

We now take a step further, and consider *directed* graphs. The directed case poses major challenges, even if just for the mere fact that directed graphs are not necessarily diagonalizable. In fact, even directed graphs with a favorable structure such as vertex-transitive graphs can be non diagonalizable [God82]. The directed Laplacian and its application were studied in, e.g., [BL92, Chu05, Bau12]. Recently, Cohen et al. [CKP+16] gave faster algorithms

[1] The fact that $\sum_i c_i \mathcal{S}^i = V f(\Sigma) V^\dagger$ is a theorem, see, e.g., [Hig08].

for computing fundamental quantities associated with random walks on directed graphs by improving the running time of solving directed Laplacian systems.

Any operator $\mathcal{A}$ can be represented by its singular value decomposition (SVD) $\mathcal{A} = U\Sigma V$, where U and V are unitary, and Σ is diagonal with the singular values on the diagonal. Another representation of $\mathcal{A}$ is by its *Jordan normal form*, $\mathcal{A} = V\mathbf{A}V^{-1}$, where V is a basis and $\mathbf{A}$ is the matrix of Jordan blocks. The elements on the diagonals of the Jordan blocks are the eigenvalues of $\mathcal{A}$ (with multiplicity as the multiplicity of the roots of its characteristic polynomial). The SVD is the usual representation of choice as it is stable, whereas the Jordan normal form is notoriously unstable to compute (see, e.g., [GVL96, Chapter 7], [Dem97, Chapter 4] and [Gly87]). However, the SVD representation is not convenient when considering BPL algorithms, as $\mathcal{A}$ does not share the same singular vectors with powers of $\mathcal{A}$. Thus, in this work, we choose to analyze our algorithm using the Jordan normal form. Admittedly, one should expect severe stability problems using such an approach. Surprisingly, we show that under mild conditions we manage to overcome these stability problems.

As before, we would like to approximate the generalized inverse $\mathcal{L}^\star$. There are two main issues to consider:

1. Peeling-off the 1-subspace. To do so, we need a good approximation of the stationary distribution of the corresponding random walk. In the undirected case, it can be easily inferred (i.e., in L) from the input. Here, we require it as an input to our algorithm.

2. Analyzing the convergence of the Taylor series of $(\mathcal{I} - \mathcal{A})^{-1}$ for a non diagonalizable $\mathcal{A}$. Recall that when a function f acts on a diagonalizable matrix $\mathcal{A}$, it acts on its eigenvalues in the natural way. In the non diagonalizable case, f acts on a Jordan *block*, which might have a large dimension, and although an eigenvalue λ on the diagonal is still mapped to an eigenvalue $f(\lambda)$, the structure of the rest of the block is no longer maintained, so we need to give this issue further consideration.

To address the second issue above, we use the theory of matrix functions that tells us exactly what $f(\mathbf{A})$ is. It turns out that there is a direct connection between $f(\mathbf{A})$, the dimension of the Jordan block, and the derivatives of f on the corresponding eigenvalue. Exploiting this connection, we manage to bound the number of terms in the Taylor series that is sufficient for convergence. The caveat here is that two "stability" parameters enter the picture. First, the spectral gap (whose formal definition we defer), which for directed graphs may no longer be at most polynomially-small and naturally affect the performance of our algorithm. Second, we also need the Jordan basis matrix V of $\mathcal{L}$ to be well-conditioned. We prove:

Theorem 8.1.2 (Informal)**.** *There exists a probabilistic algorithm that gets as input an $n \times n$ stochastic matrix $\mathcal{S}$, desired accuracy and confidence parameters $\varepsilon, \delta > 0$, $\gamma > 0$ which is a lower-bound on the spectral gap of $\mathcal{S}$, κ which is an upper bound on the condition number of the Jordan basis of $\mathcal{S}$, and outputs with probability at least $1 - \delta$ an approximation of $\mathcal{L}^\star = (\mathcal{I} - \mathcal{S})^\star$ to within an ε-accuracy, using*

$$O\left(\log \frac{n}{\gamma\varepsilon} + \log\log \frac{\kappa}{\delta}\right)$$

space.

Remarkably, the dependency of the space complexity on the condition number of the Jordan basis matrix is *doubly-logarithmic*. This also allows us to show our algorithm operates well on operators for which the eigenvalues are polynomially far apart (see Theorem 6.5.8).

Having this theorem we show that in addition to undirected graphs, our approximation algorithm works for well-conditioned regular and Eulerian directed graph (which we know have a non-negligible spectral gap and their stationary distribution is fully-explicit) and general well-conditioned rapidly-mixing directed graphs. We thus see that the algorithm manages to approximate the solution of Laplacian systems over a large (and natural) class of directed graphs.

Shortly after our work was published, Murtagh et al. [MRSV17] obtained a *deterministic* space-bounded algorithm for approximating Laplacian linear systems, of *undirected* graph, using a completely different approach. Inspired by Pend and Spielman's *time*-bounded algorithm for Laplacian systems [PS14], Murtagh et al. were able to approximate the Laplacian's pseudo-inverse via a clever identity used in a recursive manner, where squares of matrices were sparsified using the derandomized squaring of Rozenman and Vadhan [RV05]. Interestingly, while in [RV05] the derandomized squaring is shown to improve the graph's connectivity without a significant degree blow-up, in [MRSV17] they prove that the Laplacian of the derandomized squaring graph in fact *approximates* the Laplacian of the true square. The space complexity they get is $O(\log n \cdot \log\log(n/\varepsilon))$. An interesting challenge would be to try and "liberate" the error, namely coming up with a deterministic approximation algorithm using $O(\log n \log\log n + \log(1/\varepsilon))$ space.

8.2 Preliminaries

8.2.1 Basic facts from linear algebra

For a matrix $\mathcal{A} \in \mathbb{C}^{n\times n}$, $\mathcal{A}^\dagger$ is its conjugate transpose. When it might not be clear from the context, for a vector $v \in \mathbb{C}^n$, we denote $|v\rangle$ as the column vector and $\langle v|$ as the row vector, so $\langle u|\, v\rangle$ is a scalar and $|v\rangle\langle u|$ is a rank-one matrix.

Every matrix $\mathcal{A}$ has a *singular value decomposition* (SVD) $\mathcal{A} = U\Sigma V^\dagger$, where U and V are unitary and Σ is a diagonal matrix with non-negative entries, known as the singular values of $\mathcal{A}$.

The *spectrum* of a matrix $\mathcal{A}$, denoted $\mathsf{Spec}(\mathcal{A})$, is its set of (complex or real) eigenvalues. The *spectral radius* $\rho(\mathcal{A})$ of $\mathcal{A}$ is the largest absolute value of its eigenvalues. The *operator norm* $\|\mathcal{A}\|$ is $\max_{\|x\|_2=1} \|\mathcal{A}x\|$, which is also the largest singular value of $\mathcal{A}$. Notice that it is possible for $\|\mathcal{A}\|$ to be strictly larger than $\rho(\mathcal{A})$. The operator norm is sub-multiplicative. When $\mathcal{A}$ is invertible, $\kappa(\mathcal{A}) = \|\mathcal{A}\|\|\mathcal{A}^{-1}\|$ is its *condition number*. Also, we denote $\|\mathcal{A}\|_\infty$ as the induced ℓ_∞ norm, that is $\|\mathcal{A}\|_\infty = \max_{i\in[n]} \sum_{j\in[n]} |\mathcal{A}[i,j]|$. It holds that $\|\mathcal{A}\|_\infty \le \sqrt{n}\|\mathcal{A}\|$.

For an eigenvalue λ of $\mathcal{A}$, a λ-right-eigenvector (or simply an eigenvector with eigenvalue λ) is a vector v such that $\mathcal{A}v = \lambda v$. A λ-left-eigenvector is a vector v such that $v^\dagger\mathcal{A} = \lambda v^\dagger$. We define the spectral gap $\gamma(\mathcal{A}) = 1 - \max_{\lambda\in\mathsf{Spec}(\mathcal{A}),\lambda\neq 1} |\lambda|$. Note that $\gamma(\mathcal{A}) \le \min_{\lambda\in\mathsf{Spec}(\mathcal{A}),\lambda\neq 1} |1-\lambda|$.

We denote by $\mathbf{1}$ the column vector of all ones and similarly $\mathbf{0}$ the column vector of all zeros.

8.2.2 Simulatable families of matrices

Often we are interested in approximating a *value* (e.g., an entry in a matrix with integer values or the whole matrix) with a probabilistic machine. More precisely, assume there exists some value $u = u(x) \in \mathbb{R}$ that is determined by the input $x \in \{0,1\}^n$. We say a probabilistic TM $M(x,y)$ (ε,δ)-approximates $u(x)$ if

$$\forall_{x\in\{0,1\}^n} \quad \Pr_y \left[|M(x,y) - u(x)| \geq \varepsilon\right] \leq \delta.$$

If u is multi-valued (say, a vector) we say a TM (ε,δ)-approximates u if given an index i it (ε,δ)-approximates $u[i]$.

A random walk on a graph G (or its transition matrix $\mathcal{A}$) can be simulated by a probabilistic logspace machine. As a consequence, a probabilistic logspace machine can approximate powers of $\mathcal{A}$ well. Here we try to extend this notion and say that a matrix $\mathcal{A}$ is *simulatable* if any power of it can be approximated by a probabilistic algorithm running in small space. Formally:

Definition 8.2.1. *We say that a family of matrices $\mathcal{F}$ is* simulatable *if there exists a probabilistic algorithm that on input $\mathcal{A} \in \mathcal{F}$ of dimension n with $\|\mathcal{A}\| \leq \mathsf{poly}(n)$, $k \in \mathbb{N}$, $s,t \in [n]$, runs in space $O(\log \frac{nk}{\varepsilon\delta})$ and (ε,δ)-approximates $\mathcal{A}^k[s,t]$.* ♢

We prove:

Lemma 8.2.2. *The family of transition matrices of (either directed or undirected) graphs is simulatable.*

Proof. Let $G = (V,E)$ be a graph with n vertices and let $\mathcal{A}$ be its transition matrix. Let $k \in \mathbb{N}$, $s,t \in [n]$ and $\delta,\varepsilon > 0$. Consider the algorithm that on input k,s,t, takes T independent random walks of length k over G starting at vertex s. The algorithm outputs the ratio of walks that reach vertex t. Let Y_i be the random value that is 1 if the i-th trial reached t and 0 otherwise. Then, for every i, $\mathbb{E}[Y_i] = \mathcal{A}^k[s,t]$. Also, $Y_1,\ldots,Y_T$ are independent. By Chernoff,

$$\Pr\left[\left|\frac{1}{T}\sum_{i=1}^{T} Y_i - \mathcal{A}^k[s,t]\right| \geq \varepsilon\right] \leq 2e^{-2\varepsilon^2 T}.$$

Taking $T = \mathsf{poly}(\varepsilon^{-1}, \log\delta^{-1})$, the error probability (i.e., getting an estimate that is ε far from the correct value) is at most δ. Altogether, the algorithm runs in space

$$O(\log(Tnk|E|)) = O(\log(nk\varepsilon^{-1}) + \log\log\delta^{-1}),$$

assuming $|E| = \mathsf{poly}(n,k)$. □

One can also show that the family of matrices with induced infinity norm of at most 1 (that is, $\sum_j |\mathcal{A}[i,j]| \leq 1$ for every $i \in [n]$) is simulatable.

We will need a simple corollary of Lemma 8.2.2:

Lemma 8.2.3 ([DTS15b]). *Let $\mathcal{A} \in \mathbb{C}^{n\times n}$ be a stochastic matrix and let $p = \sum_{i=0}^{d} c_i x^i$ be a complex polynomial such that:*

- *For every i, $|c_i| \leq M$, and,*
- *The coefficients c_i are explicit in the sense that there exists an algorithm that given $k \leq d, \varepsilon, \delta$ outputs an (ε, δ)-approximation of c_k using $O(\log \frac{nMd \log \frac{1}{\delta}}{\varepsilon})$ space.*

Then, the entries of $p(\mathcal{A})$ can be (ε, δ)-approximated using $O(\log \frac{nMd \log \frac{1}{\delta}}{\varepsilon})$ space.

8.2.3 The Perron-Frobenius theorem

The *underlying graph* of a matrix $\mathcal{A}$ has an edge (i, j) iff $\mathcal{A}[i, j] \neq 0$. A matrix $\mathcal{A}$ is *irreducible* if its underlying directed graph is strongly connected. When $\mathcal{A}$ is irreducible, its *period* is the greatest common divisor of the lengths of the closed directed paths in the underlying directed graph of $\mathcal{A}$. We say that $\mathcal{A}$ is *aperiodic* if its period is 1. A matrix $\mathcal{A}$ is *non-negative* if all its entries are non-negative, and it is *stochastic* if it is non-negative and every row sums to 1. We will need the Perron-Frobenius theorem for irreducible non-negative matrices (see, e.g., [GR13, Chapter 8]).

Theorem 8.2.4. *Let $\mathcal{A}$ be an irreducible non-negative $n \times n$ matrix with period h and spectral radius $\rho(A) = r$. Then:*

1. *There exists an r-right-eigenvector v_1 and an r-left-eigenvector u_1 whose components are all positive.*
2. *$\mathcal{A}$ has exactly h complex eigenvalues with absolute value r and each one of them is a product of r with a different h-th root of unity. Consequently, if $\mathcal{A}$ is aperiodic then r is a simple eigenvalue, and all other eigenvalues have absolute value strictly smaller than r.*
3. *It holds that $\lim_{k\to\infty} \mathcal{A}^k / r^k = |v_1\rangle\langle u_1|$, where v_1 and u_1 are normalized so that $\langle v_1 | u_1 \rangle = 1$.*

If $\mathcal{A}$ is stochastic then $r = 1$. Furthermore, if $\mathcal{A}$ is stochastic, irreducible and aperiodic then v_1 is the all-ones vector $\mathbf{1}$ and $u_1 = \pi$ is the stationary distribution *of the corresponding random walk (all up to normalizations).*

8.2.4 Jordan normal form

Fact 8.2.5. *Every complex $n \times n$ matrix $\mathcal{A}$ can be expressed in a Jordan normal form $\mathcal{A} = V\mathbf{A}V^{-1}$ where $\mathbf{A} = \mathsf{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_B)$,*

$$\mathbf{A}_b = \mathbf{A}_b(\lambda_b) = \begin{pmatrix} \lambda_b & 1 & & \\ & \lambda_b & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_b \end{pmatrix} \in \mathbb{C}^{\dim_b \times \dim_b},$$

and $\dim_1 + \ldots + \dim_b = n$. *The Jordan matrix* $\mathbf{A}$ *has the eigenvalues of* $\mathcal{A}$ *on its diagonal, and is unique up to the ordering of the blocks* $\mathbf{A}_b$. *For an eigenvalue* λ_b, *its algebraic multiplicity is the number of times it appears on the diagonal* $\mathbf{A}$ *and its geometric multiplicity is the number of blocks having* λ_b *on their diagonal. We say an eigenvalue is* simple *if its algebraic multiplicity is one.*

Claim 8.2.6 ([Bro88], Chapter 3)**.** *Let* $\mathcal{A}$ *be an* $n \times n$ *complex matrix and let* $\mathcal{A} = V\mathbf{A}V^{-1}$ *be the Jordan normal form of* $\mathcal{A}$, *where* $\mathbf{A} = \mathsf{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_B)$. *Then, every Jordan block* $\mathbf{A}_b$ *corresponds to an* $\mathcal{A}$*-invariant subspace* $E_b = \mathsf{Ker}\left((\lambda_b \mathcal{I} - \mathcal{A})^{\dim_b}\right)$ *of dimension* $\dim_b$. *This gives a decomposition* $\mathbb{C}^n = \bigoplus_{b=1}^{B} E_b$.

For a Jordan decomposition $\mathcal{A} = V\mathbf{A}V^{-1}$, we will often write $\mathcal{A} = \sum_{b=1}^{B} V_b \mathbf{A}_b U_b$, where $\mathbf{A}_b$ is the b-th Jordan block, V_b are the columns of V that correspond to this block and similarly U_b are the rows of V^{-1} that correspond to this block.

When the operator is irreducible, aperiodic and stochastic, we can express the Perron-Frobenius theorem in the Jordan terminology and get:

Claim 8.2.7. *Let* $\mathcal{S}$ *be an irreducible, aperiodic and stochastic matrix with a stationary distribution* π *so that* $\langle \mathbf{1} | \pi \rangle = 1$ *and let* $\mathcal{S} = \sum_{b=1}^{B} V_b \mathbf{S}_b U_b$ *be a Jordan decomposition of* $\mathcal{S}$. *Then,*

- $\mathbf{S}_1 = (1)$, *the* 1×1 *matrix with an entry* 1.
- *For all* $b \geq 2$, $U_b V_1 = U_b |\mathbf{1}\rangle = \mathbf{0}$ *and* $U_1 V_b = \langle \pi | V_b = \mathbf{0}^\dagger$. *Also,* $\sum_{b=1}^{B} V_b U_b = \mathcal{I}$.
- $V_1 \mathbf{S}_1 U_1 = |\mathbf{1}\rangle \langle \pi|$ *so* $\mathcal{S} = |\mathbf{1}\rangle \langle \pi| + \sum_{b=2}^{B} V_b \mathbf{S}_b U_b$.

Proof. If v is a (right) eigenvector of $\mathcal{S}$ with eigenvalue λ then $v \in \mathsf{Im}(\cup_{b:\lambda_b = \lambda} V_b)$. Similarly, if w is a left eigenvector of $\mathcal{S}$, then its eigenvalue is an eigenvalue of $\mathcal{S}$ and $w \in \mathsf{Im}(\cup_{b:\lambda_b=\lambda} U_b)$ (this is because $\mathcal{A}$ and $\mathcal{A}^\dagger$ have the same spectrum, see, e.g., [Bro91, Chapter 9]).

Now, since $\mathcal{S}$ is stochastic, $\mathbf{1}$ is a 1-eigenvector. Also, there is a 1-left-eigenvector that we denote by π, and we normalize π such that $\langle \pi | \mathbf{1} \rangle = 1$. Furthermore, by the Perron-Frobenius theorem, the 1-eigenvalue is simple, so $\mathbf{S}_1 = (1)$, U_1 is a $1 \times n$ matrix and V_1 is a $n \times 1$ matrix. Furthermore, by the above, $\pi \in \mathsf{Im}(U_1)$, and since the dimension of the image is 1, we must have $\mathsf{Im}(U_1) = \mathrm{Span}(\{\pi\})$. Similarly, $\mathsf{Im}(V_1) = \mathrm{Span}(\{\mathbf{1}\})$. This completes the proof of the first item.

For the second item, let $U = V^{-1}$ and observe that since $UV = \mathcal{I}$, $\langle u_i | v_j \rangle = \delta_{i,j}$ (where u_i is the i-th row of U and v_j is the j-th column of V). Now, consider $b \neq b'$ and the product $P = U_b V_{b'}$. Every entry of P is of the form $\langle u_{b,i} | v_{b',j} \rangle$ where $i \in [\dim_b]$ and $j \in [\dim_{b'}]$. By the previous observation, they are all zeros. Also, $\mathcal{I}$ has a Jordan decomposition $V\mathbf{I}U$, so immediately it is clear that $\sum_{b=1}^{B} V_b U_b = \mathcal{I}$.

For the third item, Suppose $V_1 = \alpha \mathbf{1}$ and $U_1 = \beta \langle \pi |$ for some nonzero $\alpha, \beta \in \mathbb{C}$. We see that $V_1 \mathbf{S}_1 U_1 = \alpha\beta |\mathbf{1}\rangle\langle\pi|$. We want to determine $\alpha\beta$. Since $\langle \pi | \mathcal{S} = \langle \pi |$ we have that

$$\langle\pi| = \langle\pi|\,\mathcal{S} = \beta^{-1}U_1\mathcal{S} = \beta^{-1}U_1\sum_{b=1}^{B} V_b\mathbf{S}_bU_b$$

$$= \beta^{-1}U_1V_1\mathbf{I}_1U_1 + \beta^{-1}\sum_{b=2}^{B} U_1V_b\mathbf{S}_bU_b = \beta^{-1}\beta\alpha\beta\,\langle\pi|\,\mathbf{1}\rangle\,\langle\pi| \;=\; \alpha\beta\,\langle\pi|\,,$$

so $\alpha\beta = 1$. Hence, $V_1\mathbf{S}_1U_1 = V_1U_1 = |\mathbf{1}\rangle\langle\pi|$. □

8.2.5 Functions of matrices

This subsection follows the book of Higham [Hig08]. In the Jordan basis, each Jordan block is a matrix with some complex value λ over the main diagonal and 1 in the diagonal above it. We want to distinguish upper triangular matrices in which elements on the same diagonal have the same value. We note that this class D of matrices is closed under matrix addition and multiplication. We denote:

Definition 8.2.8. *For $0 \le i \le n-1$ let $\mathcal{D}_{n,i}$ be the $n \times n$ matrix that has 1 over the i-th diagonal and 0 elsewhere, where the 0-th diagonal is the main diagonal and the i-th diagonal is the diagonal i elements above it.* ◊

Clearly $D = \mathrm{Span}\,\{\mathcal{D}_{n,0}, \ldots, \mathcal{D}_{n,n-1}\}$ is closed under matrix addition. Also, since

$$\mathcal{D}_{n,i} \cdot \mathcal{D}_{n,j} \;=\; \mathcal{D}_{n,i+j},$$

D is also closed under matrix multiplication.

Suppose $p \in \mathbb{C}[x]$ is a polynomial $p(x) = \sum_{i=0}^{d} c_i x^i$. We can evaluate the polynomial over the ring $M_n(\mathbb{C})$, i.e., given an $n \times n$ matrix $\mathcal{A}$ we let

$$p(\mathcal{A}) = \sum_{i=0}^{d} c_i\mathcal{A}^i.$$

Note that if $\mathcal{A} = V\mathbf{A}V^{-1}$ then $p(\mathcal{A}) = Vp(\mathbf{A})V^{-1}$. Also, if $\mathbf{A} = \mathsf{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_B)$ then $p(\mathbf{A}) = \mathsf{diag}(p(\mathbf{A}_1), \ldots, p(\mathbf{A}_B))$. In the extreme case where $\mathcal{A}$ is diagonalizable and all Jordan blocks have dimension 1, we see that p acts on the eigenvalues of $\mathcal{A}$. In the general case, we need to understand how p acts on a Jordan block $\mathbf{A}_b = \lambda_b\mathbf{I} + \mathcal{D}_{\dim_b,1}$. The answer is quite surprising and holds for arbitrary differentiable functions.

Lemma 8.2.9 ([Hig08], Chapter 1)**.** *Let $f\colon \mathbb{C} \to \mathbb{C}$ and suppose it is differentiable n times on $\mathsf{Spec}(\mathcal{A})$. Let $\mathbf{A} \in \mathbb{C}^{n\times n}$ be a Jordan block $\mathbf{A} = \lambda\mathbf{I} + \mathcal{D}_{n,1}$. Then,*

$$f(\mathbf{A}) \;=\; \begin{pmatrix} f(\lambda) & f'(\lambda) & \ldots & \frac{f^{(n-1)}(\lambda)}{(n-1)!} \\ & f(\lambda) & \ddots & \vdots \\ & & \ddots & f'(\lambda) \\ & & & f(\lambda) \end{pmatrix} \;=\; \sum_{t=0}^{n-1} \frac{f^{(t)}(\lambda)}{t!}\mathcal{D}_{n,t}.$$

8.2.6 The generalized inverse

Let $\mathcal{A}$ be any complex linear operator. A generalized (reflexive) inverse $\mathcal{A}^+$ of $\mathcal{A}$ is a matrix that satisfies both $\mathcal{A}\mathcal{A}^+\mathcal{A} = \mathcal{A}$ and $\mathcal{A}^+\mathcal{A}\mathcal{A}^+ = \mathcal{A}^+$. A generalized inverse is not unique, however if we further demand that both $\mathcal{A}\mathcal{A}^+$ and $\mathcal{A}^+\mathcal{A}$ are Hermitian, then such an operator is unique, and is called the Moore-Penrose pseudo-inverse and can be computed using the singular values decomposition (SVD). If $\mathcal{A} = U\Sigma V^\dagger$ is the SVD of $\mathcal{A}$ then the pseudo-inverse is $\mathcal{A}^+ = V\Sigma^+U^\dagger$ where $\Sigma^+ = \mathsf{inv}(\Sigma)$ and $\mathsf{inv}(x)$ is the univariate function that is $1/x$ when $x \neq 0$ and 0 otherwise.

We will not work with the SVD but rather with the Jordan canonical form. Let $\mathcal{A} = V\mathbf{A}V^{-1}$ be a Jordan decomposition of a singular matrix $\mathcal{A}$. When the algebraic multiplicity of the eigenvalue 0 is one, the matrix $\mathcal{A}^\star = \mathsf{inv}(A)$, according to Subsection 8.2.5, is well defined. Namely, $\mathsf{inv}(A) = V\mathbf{A}^{\mathsf{inv}}V^{-1}$ where $\mathbf{A}^{\mathsf{inv}}$ is obtained by inverting every Jordan block that does not correspond to the zero eigenvalue. It is immediate that $\mathcal{A}^\star$ is a generalized inverse, although it does not generally coincide with the pseudo-inverse. From here onward, we denote $\mathcal{A}^\star$ as the generalized inverse $\mathsf{inv}(A)$.

Any generalized inverse $\mathcal{A}^\star$ can be used to determine if a system of linear equations has any solution (and if so, to give them all). More concretely, if the system $\mathcal{A}x = b$ has a solution then all its solution are given by $x = \mathcal{A}^\star b + (I - \mathcal{A}^\star\mathcal{A})w$ for an arbitrary w. All of the above claims can be found, e.g., in [BIG03].

It will later be evident that when $\mathcal{A} = \mathcal{L} = \mathcal{I} - \mathcal{S}$ is a Laplacian corresponding to an irreducible, aperiodic and stochastic matrix $\mathcal{S}$ with a stationary distribution π, the expression $\mathcal{I} - \mathcal{A}^\star\mathcal{A}$ is simply $|\mathbf{1}\rangle\langle\pi|$. Thus, if we find $\mathcal{L}^\star$ we can solve any set of equations $\mathcal{L}x = b$ that has a solution. In fact, this also works when we try to solve the system $\mathcal{L}x = b$ for b that does not admit any perfect solution, but is close to a vector in $\mathsf{Im}(\mathcal{L})$. To see that, say b is arbitrary, and on input b and $\mathcal{L}$ we output $z = \mathcal{L}^\star b$. Then $\|\mathcal{L}z - b\| = \|(\mathcal{L}\mathcal{L}^\star - \mathcal{I})b\| = \||\mathbf{1}\rangle\langle\pi|\, b\| = \sqrt{n}\cdot|\langle\pi, b\rangle|$, and so if b is δ-close to being perpendicular to π (and so close to being in $\mathsf{Im}(\mathcal{L})$) then the solution $z = \mathcal{L}^\star b$ is such that $\mathcal{L}z$ is $\sqrt{n}\delta$-close to the desired value b.

8.3 Approximating $(\mathcal{I} - \mathcal{A})^{-1}$

We start with the simple case of normal matrices, and consider general functions.

Theorem 8.3.1. *Let $f, p\colon \mathbb{C} \to \mathbb{C}$ and $\varepsilon > 0$. Suppose $\mathcal{A}$ is a normal matrix such that for every $\lambda \in \mathsf{Spec}(A)$, $|f(\lambda) - p(\lambda)| \leq \varepsilon$. Then, $\|f(\mathcal{A}) - p(\mathcal{A})\| \leq \varepsilon$.*

Proof. $\mathcal{A}$ is normal, so it is diagonalizable by a unitary matrix, $\mathcal{A} = UDU^\dagger$. Also, $f(\mathcal{A}) = Uf(D)U^\dagger$ and $p(\mathcal{A}) = Up(D)U^\dagger$. Thus, we have that

$$\|f(\mathcal{A}) - p(\mathcal{A})\| \;\leq\; \|U\|\|U^\dagger\|\|f(D) - p(D)\| \;=\; \|f(D) - p(D)\|,$$

and $\|f(D) - p(D)\|$ is simply $\max_{\lambda\in\mathsf{Spec}(\mathcal{A})}|f(\lambda) - p(\lambda)| \leq \varepsilon$. □

With that we can easily see that when $\mathcal{A}$ is normal, $\sum_{i=0}^{T}\mathcal{A}^i$ approximates $(\mathcal{I} - \mathcal{A})^{-1}$ pretty well. Formally,

Corollary 8.3.2. *Let $\mathcal{A}$ be a normal matrix and suppose $\mathsf{Spec}(\mathcal{A}) \subseteq [0,1)$ and in particular $\mathcal{I} - \mathcal{A}$ is invertible. Then,*

$$\|(\mathcal{I}-\mathcal{A})^{-1} - \sum_{i=0}^{T} \mathcal{A}^i\| \;\leq\; \frac{e^{-T\overline{\lambda}(\mathcal{A})}}{\overline{\lambda}(\mathcal{A})}.$$

Proof. For $\lambda \in [0,1)$, it holds that

$$\left|\frac{1}{1-\lambda} - \sum_{i=0}^{T} \lambda^i\right| \;\leq\; \sum_{T+1}^{\infty} \lambda^i \;=\; \frac{\lambda^{T+1}}{1-\lambda}.$$

The above expression is maximized where $\lambda = 1-\gamma(\mathcal{A})$, so we have:

$$\left|\frac{1}{1-\lambda} - \sum_{i=0}^{T} \lambda^i\right| \;\leq\; \frac{(1-\gamma(\mathcal{A}))^T}{\gamma(\mathcal{A})} \;\leq\; \frac{e^{-T\gamma(\mathcal{A})}}{\gamma(\mathcal{A})},$$

and the corollary follows. □

We would like to extend this result to arbitrary operators $\mathcal{A}$. As a first attempt we begin with generalizing Theorem 8.3.1 to arbitrary operators. For that we need the representation of $\mathcal{A}$ in its Jordan normal form, and we also need the function p and its derivatives to approximate the target function f and its derivatives well. We prove:

Theorem 8.3.3. *Let $f, p\colon \mathbb{C} \to \mathbb{C}$. Suppose $\mathcal{A}$ is an $n \times n$ matrix such that for every $\lambda \in \mathsf{Spec}(A)$ and every $k \leq n$, $|f^{(k)}(\lambda) - p^{(k)}(\lambda)| \leq k! \cdot \varepsilon_k$. Furthermore, assume $\mathcal{A}$ has a Jordan decomposition $\mathcal{A} = V\mathbf{A}V^{-1}$, and the largest Jordan block has dimension D. Then, $\|f(\mathcal{A}) - p(\mathcal{A})\| \leq \kappa(V) \cdot \sum_{k=0}^{D-1} \varepsilon_k$.*

Proof. Let $\mathbf{A} = \mathbf{A}_1 \oplus \ldots \oplus \mathbf{A}_b$, corresponding to the different Jordan blocks. By Lemma 8.2.9, $f(\mathcal{A}) = Vf(\mathbf{A})V^{-1}$ where $f(\mathbf{A}) = f(\mathbf{A}_1) \oplus \ldots \oplus f(\mathbf{A}_b)$,

$$f(\mathbf{A}_i) \;=\; \begin{pmatrix} f(\lambda_i) & f'(\lambda_i) & \ldots & \frac{f^{(\dim_i - 1)}(\lambda_i)}{(\dim_i - 1)!} \\ & f(\lambda_i) & \ddots & \vdots \\ & & \ddots & f'(\lambda_i) \\ & & & f(\lambda_i) \end{pmatrix} \;=\; \sum_{k=0}^{\dim_i - 1} \frac{f^{(k)}(\lambda_i)}{k!} \mathcal{D}_{\dim_i, k},$$

and λ_i is the eigenvalue corresponding to the block $\mathbf{A}_i$ of dimension $\dim_i$. The same of course holds for p. Thus,

$$\|f(\mathcal{A}) - p(\mathcal{A})\| \;=\; \|V(f(\mathbf{A}) - p(\mathbf{A}))V^{-1}\| \;\leq\; \kappa(V) \cdot \|f(\mathbf{A}) - p(\mathbf{A})\|.$$

To bound the latter expression, note that

$$\begin{aligned} \|f(\mathbf{A}) - p(\mathbf{A})\| &= \max_{i \in [b]} \|f(\mathbf{A}_i) - p(\mathbf{A}_i)\| \\ &\leq \max_{i \in [b]} \sum_{k=0}^{\dim_i - 1} \left|\frac{f^{(k)}(\lambda_i) - p^{(k)}(\lambda_i)}{k!}\right| \|\mathcal{D}_{\dim_i, k}\| \;\leq\; \sum_{k=0}^{D-1} \varepsilon_k. \end{aligned}$$

□

When $\mathcal{A}$ is normal, $\kappa(V) = 1$ and the maximal block length is 1, so we recover Theorem 8.3.1. We now check what we get for $(\mathcal{I} - \mathcal{A})^{-1}$ and an arbitrary operator $\mathcal{A}$:

Corollary 8.3.4. *Suppose $\mathcal{A}$ is an $n \times n$ matrix that has a Jordan decomposition $\mathcal{A} = V\mathbf{A}V^{-1}$. Suppose every eigenvalue λ of $\mathcal{A}$ satisfies $|\lambda| < 1$ and in particular $\mathcal{I} - \mathcal{A}$ is invertible. Let $T \in \mathbb{N}$ such that $T \geq \frac{8n^2}{\gamma(\mathcal{A})^2}$, let $f(\mathcal{A}) = (\mathcal{I} - \mathcal{A})^{-1}$ and $p(\mathcal{A}) = \sum_{i=0}^{T} \mathcal{A}^i$. Then,*

$$\|f(\mathcal{A}) - p(\mathcal{A})\| \;\leq\; 2n\kappa(V)\frac{e^{-T\gamma(\mathcal{A})/4}}{\gamma(\mathcal{A})}.$$

Proof. Let $\mathcal{A}$ be an $n \times n$ matrix and suppose every eigenvalue λ of $\mathcal{A}$ satisfies $|\lambda| < 1$. We consider, again, inverting $\mathcal{I} - \mathcal{A}$ by considering the function $f(\lambda) = \frac{1}{1-\lambda}$ and its power-series expansion $p(\lambda) = \sum_{i=0}^{T} \lambda^i$. For $k \leq n$, one can verify that $\frac{1}{k!}f^{(k)}(\lambda) = \frac{1}{(1-\lambda)^{k+1}}$ and $\frac{1}{k!}p^{(k)}(\lambda) = \sum_{i=0}^{T-k}\binom{k+i}{k}\lambda^i$. Also, $\frac{1}{k!}f^{(k)}(\lambda) = \sum_{i=0}^{\infty}\binom{k+i}{k}\lambda^i$ so we see that

$$\varepsilon_k \;=\; \left|\sum_{i=T-k+1}^{\infty}\binom{k+i}{k}\lambda^i\right|.$$

As $T \geq 4n$, $T - k + 1 \geq T/2$. Also, $\binom{k+i}{k} \leq (k+i)^k \leq (2i)^k$, so $\varepsilon_k \;\leq\; \sum_{i=T/2}^{\infty}(2i)^k\lambda^i$. Now, we have that $(2i)^k \leq |\lambda|^{-i/2}$, since

$$\begin{aligned}(2i)^k|\lambda|^{i/2} \;&=\; e^{k\ln(2i)-(i/2)\ln\frac{1}{|\lambda|}} \;=\; e^{\frac{1}{2}\left(2k\ln(2i)-i\ln\frac{1}{|\lambda|}\right)} \;\leq\; e^{\frac{1}{2}\left(n\sqrt{i}-i\ln\frac{1}{|\lambda|}\right)}\\ &\leq\; e^{\frac{\sqrt{i}}{2}\left(n-\sqrt{i}\ln\frac{1}{|\lambda|}\right)} \;\leq\; e^{\frac{\sqrt{i}}{2}\left(n-\sqrt{T/2}\cdot\ln\frac{1}{1-\gamma(\mathcal{A})}\right)} \;\leq\; e^{\frac{\sqrt{i}}{2}\left(n-\sqrt{T/2}\cdot\gamma(\mathcal{A})\right)}\\ &\leq\; e^{\frac{\sqrt{i}}{2}(n-2n)} \;\leq\; 1.\end{aligned}$$

Plugging it to the above bound for ε_k, we obtain:

$$\varepsilon_k \;\leq\; \left|\sum_{i=T/2}^{\infty}\lambda^{i/2}\right| \;=\; \left|\frac{\lambda^{T/4}}{1-\sqrt{\lambda}}\right|.$$

To bound $\left|\frac{1}{1-\sqrt{\lambda}}\right|$, we use the fact that:

$$\left|\frac{1}{1-\sqrt{\lambda}}\right| \;=\; \frac{|1+\sqrt{\lambda}|}{|1-\lambda|} \;\leq\; \frac{2}{\gamma(\mathcal{A})}.$$

Altogether,

$$\varepsilon_k \;\leq\; \frac{2}{\gamma(\mathcal{A})}(1-\gamma(\mathcal{A}))^{T/4} \;\leq\; \frac{2e^{-T\gamma(\mathcal{A})/4}}{\gamma(\mathcal{A})}.$$

The Corollary follows by applying Theorem 8.3.3 and using the fact that $D \leq n$. □

8.4 Computing the Generalized Inverse

In this section we approximate the generalized inverse of the Laplacian of directed graphs as long as we have a good approximation of its stationary distribution. Formally,

Theorem 8.4.1. *There exists a probabilistic algorithm that gets as input:*

- *An $n \times n$ irreducible, aperiodic stochastic matrix $\mathcal{S}$,*
- *Two parameters, κ and γ, which describe how stable the input $\mathcal{S}$ is:*
  - *Suppose $\kappa \geq \kappa(V)$, where $\mathcal{S} = V\mathbf{S}V^{-1}$ is any Jordan decomposition of $\mathcal{S}$, and,*
  - *$\gamma(\mathcal{S}) \geq \gamma$.*
- *Desired accuracy and confidence parameters $\varepsilon, \delta > 0$.*
- *An approximation $\tilde{\pi}$ of the stationary distribution π of $\mathcal{S}$, where $\|\tilde{\pi} - \pi\| \leq \tau$ and $\tau \leq \frac{\varepsilon}{(T+1)\sqrt{n}}$ for $T = \frac{8n^2}{\gamma^2}\left(1 + \log\frac{n\kappa}{\varepsilon\gamma}\right)$.*

Let $\mathcal{L}$ denote the Laplacian $\mathcal{L} = \mathcal{I} - \mathcal{S}$. The algorithm outputs a $(3\varepsilon, \delta)$-approximation of $\mathcal{L}^\star$ using

$$O\left(\log\frac{n}{\gamma\varepsilon} + \log\log\frac{\kappa}{\delta}\right)$$

space.

Intuitively, we would like to employ the following approach. Given a stochastic operator $\mathcal{S}$ with a unique stationary distribution π, we would like to "peel off" the 1×1 Jordan block with eigenvalue 1, so that we are left with an operator $\mathcal{A}$ such that $\mathcal{I} - \mathcal{A}$ is invertible. Then, we would like to use Corollary 8.3.4 to approximate $(\mathcal{I} - \mathcal{A})^{-1}$ by $\sum_{i=0}^{T} \mathcal{A}^i$, using the fact that we can approximate $\mathcal{A}^i$ well with a BPL algorithm.

There are two obstacles that we need to overcome:

- First, when $\mathcal{S}$ in not normal, we do not have an orthonormal basis, so we need to explain what "peeling off" the stationary distribution means. It turns out that $\mathcal{A} = \mathcal{S} - |\mathbf{1}\rangle\langle\pi|$.
- Second, while $\mathcal{S}$ is stochastic, $\mathcal{A} = \mathcal{S} - |\mathbf{1}\rangle\langle\pi|$ is not, and furthermore, its ℓ_∞ norm is usually greater than 1. In particular, we cannot immediately assume that we can approximate high powers of it in BPL. We will show that $\mathcal{A}$ is still simulatable because $|\mathbf{1}\rangle\langle\pi|$ commutes with both $\mathcal{S}$ and $\mathcal{A}$.

We also need to check that the fact that $\tilde{\pi}$ is only close to π and not exactly it, does not affect the parameters by too much.

We start the formal exposition with a precise description of the algorithm.

8.4.1 The algorithm

The algorithm first computes the parameter

$$T \;=\; \left\lceil \frac{8n^2}{\gamma^2}\left(1+\log\frac{n\kappa}{\varepsilon\gamma}\right)\right\rceil .$$

The algorithm then computes an (ε,δ)-approximation of the matrix

$$\widetilde{Q}_T(\mathcal{S}) \;=\; \left(\sum_{i=0}^{T}\mathcal{S}^i\right) - (T+1)\,|\mathbf{1}\rangle\langle\tilde{\pi}|$$

using Lemma 8.2.3 (note that since $\tilde{\pi}$ is given, we *approximate* the power series and compute $(T+1)\,|\mathbf{1}\rangle\langle\tilde{\pi}|$ exactly).

We first argue that the algorithm runs in small space and then analyze correctness.

8.4.2 Efficiency

We observe:

Lemma 8.4.2. *For every $\varepsilon,\delta>0$ and integer T, and any $n\times n$ stochastic matrix $\mathcal{S}$, the entries of $\widetilde{Q}_T(\mathcal{S})$ can be (ε,δ)-approximated using $O\left(\log\frac{nT\log\frac{1}{\delta}}{\varepsilon}\right)$ space.*

Proof. The claim follows directly from Lemma 8.2.3 since $\mathcal{S}$ is stochastic. □

8.4.3 Correctness

We first do the analysis in the ideal situation that $\tilde{\pi}=\pi$ and see that in this case the algorithm $(2\varepsilon,\delta)$-approximates $\mathcal{L}^\star$. We then show that when $\|\pi-\tilde{\pi}\|\le\tau$ the algorithm $(3\varepsilon,\delta)$-approximates $\mathcal{L}^\star$.

8.4.3.1 Peeling off the 1-eigenspace

Throughout the proof we use the representation of $\mathcal{S}$ guaranteed by Claim 8.2.7. Namely, $\mathcal{S}$ can be written as $\mathcal{S}=\sum_{b=1}^{B}V_b\mathbf{S}_bU_b$ where

- $\mathbf{S}_1$ is a 1×1 matrix and $\mathbf{S}_1=(1)$. Also, $V_1U_1=|\mathbf{1}\rangle\langle\pi|$ and $\langle\mathbf{1}|\,\pi\rangle=1$,
- For all $b\ge 2$, $U_b\,|\mathbf{1}\rangle=\mathbf{0}$ and $\langle\pi|\,V_b=\mathbf{0}^\dagger$, and
- $\sum_{b=1}^{B}V_bU_b=\mathcal{I}$.

Our goal is to find the generalized inverse of $\mathcal{L}=\mathcal{I}-\mathcal{S}$. As explained before, our first step is to "peel-off" from $\mathcal{S}$ the 1-eigenspace, and the correct way to do that is by annihilating the 1×1 Jordan block with eigenvalue 1. We therefore define:

$$\mathcal{A} \;=\; \mathcal{S}-|\mathbf{1}\rangle\langle\pi| .$$

We notice that $\mathcal{S}$, $\mathcal{A}$, $\mathcal{L}$ and $\mathcal{L}^\star$ share the same Jordan basis, therefore, if we express $\mathcal{S} = \sum_{b=1}^{B} U_b \mathbf{S}_b V_b$ then

$$\mathcal{L} \;=\; \sum_{b=2}^{B} V_b(\mathbf{I}_b - \mathbf{S}_b)U_b,$$

and,

$$\mathcal{A} \;=\; \sum_{b=2}^{B} V_b \mathbf{S}_b U_b.$$

We denote $\mathbf{L}_b = \mathbf{I}_b - \mathbf{S}_b$ for $b \geq 2$ (and $\mathbf{L}_1$ is the zero matrix). The big advantage of $\mathcal{A}$ over $\mathcal{S}$ is that in $\mathcal{A}$ all eigenvalues have magnitude smaller than 1, as $\mathcal{A} = \sum_{b=2}^{B} V_b \mathbf{S}_b U_b$, and therefore $\mathcal{I} - \mathcal{A}$ is invertible. We still need, however, to relate $\mathcal{L}^\star$ to $(\mathcal{I} - \mathcal{A})^{-1}$. We prove:

Lemma 8.4.3. $\mathcal{L}^\star = (\mathcal{I} - \mathcal{A})^{-1} - |\mathbf{1}\rangle\langle\pi|$.

Proof. Recall that $\mathcal{S} = |\mathbf{1}\rangle\langle\pi| + \sum_{b=2}^{B} V_b \mathbf{S}_b U_b$, $\mathcal{A} = \sum_{b=2}^{B} V_b \mathbf{S}_b U_b$ and $\mathcal{I} = \sum_{b=1}^{B} V_b U_b$. Hence,

$$\mathcal{I} - \mathcal{A} \;=\; \sum_{b=1}^{B} V_b U_b - \sum_{b=2}^{B} V_b \mathbf{S}_b U_b \;=\; V_1 U_1 + \sum_{b=2}^{B} V_b(\mathbf{I}_b - \mathbf{S}_b)U_b \;=\; |\mathbf{1}\rangle\langle\pi| + \sum_{b=2}^{B} V_b \mathbf{L}_b U_b.$$

The inverse is thus given by

$$(\mathcal{I} - \mathcal{A})^{-1} = |\mathbf{1}\rangle\langle\pi| + \sum_{b=2}^{B} V_b \mathbf{L}_b^{-1} U_b = |\mathbf{1}\rangle\langle\pi| + \mathcal{L}^\star,$$

as desired. □

Intuitively, this means that approximating $(\mathcal{I} - \mathcal{A})^{-1}$ suffices for approximating $\mathcal{L}^\star$, and we next consider approximating $(\mathcal{I} - \mathcal{A})^{-1}$.

8.4.3.2 Approximating $(\mathcal{I} - \mathcal{A})^{-1}$

Since all eigenvalues of $\mathcal{A}$ have magnitude smaller than 1, we can apply Corollary 8.3.4 and get:

Lemma 8.4.4.

$$\|(\mathcal{I} - \mathcal{A})^{-1} - \sum_{k=0}^{T} \mathcal{A}^k\| \;\leq\; \varepsilon.$$

Proof. We saw that $\mathcal{A} = \sum_{b=2}^{B} V_b \mathbf{S}_b U_b$, and by the Perron-Frobenius theorem the eigenvalues that are written on $\mathbf{S}_b$ for $b \geq 2$, are at most $1-\gamma < 1$ in absolute value. Thus, all eigenvalues of $\mathcal{A}$ have absolute value at most $\gamma(\mathcal{S})$. By Corollary 8.3.4, for $T \geq \frac{8n^2}{\gamma(\mathcal{S})^2}$,

$$\|(\mathcal{I} - \mathcal{A})^{-1} - \sum_{k=0}^{T} \mathcal{A}^k\| \;\leq\; 2n\kappa(V)\frac{e^{-T\gamma(\mathcal{S})/4}}{\gamma(\mathcal{S})}.$$

Substituting $T = \left\lceil \frac{8n^2}{\gamma(\mathcal{S})^2} \ln \frac{2n\kappa(V)}{\varepsilon\gamma(\mathcal{S})} \right\rceil$, the desired bound holds. □

Thus, the problem now reduces to simulating $\mathcal{A}^i$ in small space. As mentioned before, $\mathcal{A}$ is not stochastic and its ℓ_∞ norm is often larger than 1. However $\mathcal{A} = \mathcal{S} - |\mathbf{1}\rangle\langle\pi|$ has a very special form that conforms with the Jordan basis structure, which we now employ:

Claim 8.4.5. *The matrices $\mathcal{S}$ and $|\mathbf{1}\rangle\langle\pi|$ commute, and furthermore $\mathcal{S}\cdot|\mathbf{1}\rangle\langle\pi| = |\mathbf{1}\rangle\langle\pi|\cdot\mathcal{S} = |\mathbf{1}\rangle\langle\pi|$.*

Proof.

$$\mathcal{S}\cdot|\mathbf{1}\rangle\langle\pi| = |\mathbf{1}\rangle\langle\pi| + \sum_{b=2}^{B} V_b\mathbf{S}_b U_b\cdot|\mathbf{1}\rangle\langle\pi| = |\mathbf{1}\rangle\langle\pi|,$$

and,

$$|\mathbf{1}\rangle\langle\pi|\cdot\mathcal{S} = |\mathbf{1}\rangle\langle\pi| + \sum_{b=2}^{B} |\mathbf{1}\rangle\langle\pi|\cdot V_b\mathbf{S}_b U_b = |\mathbf{1}\rangle\langle\pi|.$$

□

Claim 8.4.6. *For every $k \geq 1$, $\mathcal{A}^k = \mathcal{S}^k - |\mathbf{1}\rangle\langle\pi|$.*

Proof. The proof is by induction on k. For $k = 1$ the claim follows by the definition. Assume the statement holds for $k \in \mathbb{N}$, and consider $\mathcal{A}^{k+1}$, so By Claim 8.4.5:

$$\begin{aligned}\mathcal{A}^{k+1} &= (\mathcal{S} - |\mathbf{1}\rangle\langle\pi|)\cdot(\mathcal{S}^k - |\mathbf{1}\rangle\langle\pi|)\\ &= \mathcal{S}^{k+1} - \mathcal{S}\cdot|\mathbf{1}\rangle\langle\pi| - |\mathbf{1}\rangle\langle\pi|\cdot\mathcal{S}^k + |\mathbf{1}\rangle\langle\pi|\mathbf{1}\rangle\langle\pi|\\ &= \mathcal{S}^{k+1} - |\mathbf{1}\rangle\langle\pi| - |\mathbf{1}\rangle\langle\pi| + |\mathbf{1}\rangle\langle\pi| \;=\; \mathcal{S}^{k+1} - |\mathbf{1}\rangle\langle\pi|.\end{aligned}$$

□

Thus, $\mathcal{A}$ is simulatable and we can approximate $(\mathcal{I} - \mathcal{A})^{-1}$ in small space.

8.4.3.3 Putting everything together

Define the *ideal* polynomial Q_T by:

$$Q_T(\mathcal{S}) \;=\; \left(\sum_{i=0}^{T}\mathcal{S}^i\right) - (T+1)\,|\mathbf{1}\rangle\langle\pi|.$$

Lemma 8.4.7. $\|\mathcal{L}^\star - Q_T(\mathcal{S})\| \leq \varepsilon$.

Proof.

$$\begin{aligned}\|\mathcal{L}^\star - Q_T(\mathcal{S})\| &= \|(\mathcal{I}-\mathcal{A})^{-1} - |\mathbf{1}\rangle\langle\pi| - Q_T(\mathcal{S})\|\\ &\leq \|\left(\sum_{i=0}^{T}\mathcal{A}^i\right) - |\mathbf{1}\rangle\langle\pi| - Q_T(\mathcal{S})\| + \varepsilon\\ &= \|\mathcal{A}^0 + \sum_{i=1}^{T}\left(\mathcal{S}^i - |\mathbf{1}\rangle\langle\pi|\right) - |\mathbf{1}\rangle\langle\pi| - Q_T(\mathcal{S})\| + \varepsilon\\ &= \|\left(\sum_{i=0}^{T}\mathcal{S}^i\right) - (T+1)\,|\mathbf{1}\rangle\langle\pi| - Q_T(\mathcal{S})\| + \varepsilon \;=\; \varepsilon.\end{aligned}$$

□

Finally, we check how the fact that $\tilde{\pi}$ is only close to π, affects our accuracy. We see that:

Claim 8.4.8. $\|\widetilde{Q}_T(\mathcal{S}) - Q_T(\mathcal{S})\| \leq \varepsilon$.

Proof. Notice that $\widetilde{Q}_T(\mathcal{S}) - Q_T(\mathcal{S}) = (T+1)\,|\mathbf{1}\rangle\langle\tilde{\pi} - \pi|$. Therefore, $\|\widetilde{Q}_T(\mathcal{S}) - Q_T(\mathcal{S})\| \leq (T+1)\cdot\|\mathbf{1}\|\cdot\|\tilde{\pi} - \pi\|$. The proof follows because $\|\mathbf{1}\| = \sqrt{n}$ and $\|\tilde{\pi} - \pi\| \leq \tau \leq \frac{\varepsilon}{\sqrt{n}(T+1)}$. □

Now, since we (ε, δ)-approximate $\widetilde{Q}_T(\mathcal{S})$, then except for probability δ what we output is ε-close to $\widetilde{Q}_T(\mathcal{S})$, and therefore it is 2ε-close to $Q_T(\mathcal{S})$ and 3ε-close to $\mathcal{L}^\star$, which completes the proof of Theorem 8.4.1.

8.5 Some Specific Families of Graphs

Ultimately, we would like to solve in BPL any set of equations $\mathcal{L}x = b$, where b is close to $\mathsf{Im}(\mathcal{L})$, and where $\mathcal{L}$ is the Laplacian of a stochastic matrix $\mathcal{S}$. Theorem 8.4.1 is a step towards this goal, but it works only when:

- $\mathcal{S}$ is irreducible, namely, its underlying graph is strongly connected,
- $\mathcal{S}$ is aperiodic,
- We can approximate well the unique stationary distribution π,
- $\gamma(\mathcal{S}) \geq \frac{1}{n^a}$ for some constant a, i.e., all eigenvalues except the largest one, are at most $1-\gamma$ in absolute value, and,
- $\kappa(V) \leq 2^{n^b}$ for some constant b, where $\mathcal{S} = V\mathbf{S}V^{-1}$ is a Jordan decomposition and $\kappa(V) = \|V\|\cdot\|V^{-1}\|$. Notice that here we may tolerate exponential $\kappa(V)$ as the space complexity dependency on κ is doubly-logarithmic.

In this section we want to examine which requirements can be relaxed. The section is organized as follows. First, we note that we can get rid of the aperiodicity requirement and we can somewhat relax the spectral gap requirement. Then we show that in some cases we can get rid of the $\kappa(V)$ requirement (when the eigenvalues are polynomially separated). Finally, we give specific results for:

- Undirected graphs,
- Directed Eulerian graphs (which generalize directed regular graphs), and,
- Directed rapidly-mixing graphs.

8.5.1 Omitting the aperiodicity requirement using lazy walks

Given a stochastic matrix $\mathcal{S}$ we can convert it to the corresponding lazy walk $\mathcal{S}' = \frac{1}{2}(\mathcal{I}+\mathcal{S})$, that stays in place with probability half. Define:

$$\gamma'(\mathcal{S}) \;=\; \max_{\lambda\in\mathsf{Spec}(\mathcal{S}'),\lambda\neq 1}(1-\Re(\lambda)).$$

The conversion has two benefits. First, the walk is clearly aperiodic. Also, we will be able to replace the condition $\gamma \leq \gamma(S)$, with the milder condition $\gamma \leq \gamma'(\mathcal{S})$. We will also show that we can recover the generalized inverse of the Laplacian of a graph G from that of the lazy walk variant of G. We prove:

Theorem 8.5.1. *There exists a probabilistic algorithm that gets as input:*

- *An $n \times n$ irreducible, stochastic matrix $\mathcal{S}$,*
- *Two parameters, κ and γ, which describe how stable the input $\mathcal{S}$ is:*
  - *Suppose $\kappa \geq \kappa(V)$, where $\mathcal{S} = V\mathbf{S}V^{-1}$ is any Jordan decomposition of $\mathcal{S}$, and,*
  - *$\gamma'(\mathcal{S}) \geq \gamma$.*
- *Desired accuracy and confidence parameters $\varepsilon, \delta > 0$.*
- *An approximation $\tilde{\pi}$ of the stationary distribution π of $\mathcal{S}$, where $\|\tilde{\pi}-\pi\| \leq \tau$ and $\tau \leq \frac{\varepsilon}{(T+1)\sqrt{n}}$ for $T = \frac{8n^2}{\gamma^2}\left(1+\log\frac{n\kappa}{\varepsilon\gamma}\right)$.*

Let $\mathcal{L}$ denote the Laplacian $\mathcal{L} = \mathcal{I}-\mathcal{S}$. The algorithm outputs a $(3\varepsilon,\delta)$-approximation of $\mathcal{L}^\star$ using

$$O\left(\log\frac{n}{\gamma\varepsilon} + \log\log\frac{\kappa}{\delta}\right)$$

space.

Proof. We run the algorithm of Theorem 8.4.1 over $\mathcal{S}' = \frac{1}{2}(\mathcal{I}+\mathcal{S})$. It is clear that $\mathcal{S}'$ is stochastic and aperiodic. By assumption, $\mathcal{S}'$ is irreducible (since $\mathcal{S}$ is). Also, $\mathcal{S}$ and $\mathcal{S}'$ have the same V and by assumption $\kappa(V) \leq \kappa$. They also share the same stationary distribution π, and we are given π' which is close to π.

We will soon prove that $\gamma(\mathcal{S}') \geq \frac{\gamma'(\mathcal{S})}{4}$. Therefore, by Theorem 8.4.1, we get a $(3\varepsilon,\delta)$-approximation of $(\mathcal{I}-\mathcal{S}')^\star$. Finally, we will see that $(\mathcal{I}-\mathcal{S}')^\star = 2(\mathcal{I}-\mathcal{S})^\star$ and so we easily get an approximation for $(\mathcal{I}-\mathcal{S})^\star$.

To see that indeed $(\mathcal{I}-\mathcal{S}')^\star = 2(\mathcal{I}-\mathcal{S})^\star$, notice that $\mathcal{I}$ and $\mathcal{S}$ share the same Jordan basis V. The first block in $\mathcal{S}'$ and $\mathcal{S}$ is the same, and for $b \geq 2$, if the b-th block in $\mathcal{S}$ is $\mathbf{S}_b$, then the b-th block in $(\mathcal{I}-\mathcal{S}')^\star$ is $(\mathcal{I}-\frac{1}{2}(I+\mathbf{S}_b))^{-1} = 2(I-\mathbf{S}_b)^{-1}$ and the b-th block of $(\mathcal{I}-\mathcal{S})^\star$ is $(\mathbf{I}-\mathbf{S}_b)^{-1}$.

Thus, all that is left is to prove:

Claim 8.5.2. *It holds that $\gamma(\mathcal{S}') \geq \frac{\gamma'(\mathcal{S})}{4}$.*

Proof. Fix $\lambda \in \mathsf{Spec}(\mathcal{S})$, $|\lambda| \leq 1$, and write $\lambda = a + bi$ for $a, b \in \mathbb{R}$. Also, let $\lambda' = \frac{1}{2} + \frac{1}{2}\lambda = \frac{1+a}{2} + \frac{b}{2}i$, which is the corresponding eigenvalue in $\mathcal{S}'$. Thus:

$$|\lambda'|^2 \;=\; \frac{a^2 + b^2 + 2a + 1}{4} \;\leq\; \frac{1 + 2a + 1}{4} \;=\; \frac{1 + \Re(\lambda)}{2},$$

so $1 - |\lambda'| \leq 1 - \sqrt{\frac{1+\Re(\lambda)}{2}}$. The claim follows since for every R such that $|R| \leq 1$, $1 - \sqrt{\frac{1+R}{2}} \geq \frac{1}{4}(1 - R)$. □

□

8.5.2 Undirected graphs

Given an undirected graph we can easily partition it to its connected components using the fact that st-connectivity of undirected graphs is in BPL [AKL+79] (in fact, Reingold showed it is in L [Rei08]). Therefore, we can solve the system of equations on each connected component separately.

Now, say we are given an undirected graph G and $\mathcal{A}$ is its adjacency matrix. The stochastic matrix $\mathcal{S}$ associated with G is $D^{-1}\mathcal{A}$, where D is a diagonal matrix with the degree $\deg_i$ of the i-th vertex on the i-th element of the diagonal. While $\mathcal{A}$ is Hermitian, $\mathcal{S}$ is usually not. Still, $\mathcal{S}$ is similar to a Hermitian matrix in the following form: Express $D^{-1/2}\mathcal{A}D^{-1/2} = V\mathbf{A}V^{-1}$ where V is unitary and $\mathbf{A}$ diagonal with real entries (because $D^{-1/2}\mathcal{A}D^{-1/2}$ is Hermitian), then $\mathcal{S} = (D^{-1/2}V)\mathbf{A}(D^{-1/2}V)^{-1}$. Thus, $\mathcal{S}$ has Jordan normal form $W\mathbf{A}W^{-1}$ with $W = D^{-1/2}V$. We see that

$$\begin{aligned}\kappa(W) &= \|D^{-1/2}V\| \cdot \|VD^{1/2}\| \leq \|D^{-1/2}\|\|D^{1/2}\|\|V\|\|V^{-1}\| \\ &\leq \sqrt{\frac{\lambda_{\max}(D)}{\lambda_{\min}(D)}} \leq \sqrt{\frac{n}{1}} = \sqrt{n}.\end{aligned}$$

We can therefore always take $\kappa = \sqrt{n}$ in Theorem 8.5.1 when we deal with undirected graphs, even when the graph is irregular.

The above discussion shows that $\mathcal{S}$ is similar to the diagonal matrix $\mathbf{A}$ which has a set of real eigenvalues, and therefore so does $\mathcal{S}$. Chung proved that:

Lemma 8.5.3 ([Chu97], Lemma 1.9)**.** *Let $\mathcal{S}$ be a transition matrix of an undirected connected graph with diameter Γ. Then $\gamma'(\mathcal{S}) \geq \frac{1}{\Gamma \cdot \sum_i \deg_i}$.*

Finally, we need the stationary distribution π. However, for an undirected graph $G = (V, E)$ the stationary distribution π is fully explicit and gives weight $\frac{2\deg_i}{|E|}$ to the vertex i. Altogether, we get the theorem for undirected graphs that was stated in the introduction:

Theorem 8.5.4. *There exists a probabilistic algorithm that gets as input an $n \times n$ stochastic matrix $\mathcal{S}$ that is the transition matrix of an undirected graph and desired accuracy and confidence parameters $\varepsilon, \delta > 0$, outputs a (ε, δ)-approximation of $\mathcal{L}^\star = (\mathcal{I} - \mathcal{S})^\star$ using*

$$O\left(\log \frac{n}{\varepsilon} + \log\log \frac{1}{\delta}\right)$$

space.

We note that the above theorem also holds for *weighted* undirected graphs. To see this, view $\deg_i$ as the sum of weights of the i-th vertex, $\deg_i = \sum_j \mathcal{A}[i,j]$, which is also $\lambda_i(D)$. Then, we can take $\kappa = \sqrt{\lambda_{\max}(D)/\lambda_{\min}(D)}$ in Theorem 8.5.1. The stationary distribution is again fully explicit. Finally, analogues of Lemma 8.5.3 for weighted undirected graph show that $\gamma'(\mathcal{S})$ is at least inverse-polynomially large in the weights of the graph (e.g., Section 5 in [Chu96]).

When G is undirected we can also approximate in BPL the often used *symmetric normalized Laplacian* , which is

$$\mathcal{L}^{\mathsf{sym}} = \mathcal{I} - D^{-1/2}\mathcal{A}D^{-1/2},$$

where $\mathcal{A}$ is the graph's adjacency matrix and D is the diagonal degrees matrix. We have seen that we can approximate $\mathcal{L}^\star = (\mathcal{I} - D^{-1}\mathcal{A})^\star$ in BPL, and

$$(\mathcal{L}^{\mathsf{sym}})^\star = \left(D^{1/2}\mathcal{L}D^{-1/2}\right)^\star = D^{1/2}\mathcal{L}^\star D^{-1/2}.$$

8.5.3 On the parameter $\kappa(V)$

Our algorithm's space complexity has a doubly-logarithmic dependency on $\kappa(V)$ – the minimal condition number of all Jordan bases. When the matrix $\mathcal{S}$ has well-separated eigenvalues (namely, the minimal distance between every two eigenvalues is at least polynomially-small), the dependency can be omitted. This is implied by the following theorem:

Theorem 8.5.5 ([Smi67])**.** *Let $\mathcal{A}$ be an $n \times n$ matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$ and suppose $\Delta > 0$ is such that $\min_{i \neq j} |\lambda_i - \lambda_j| \geq \Delta$. Also, let κ_A be the minimal value of $\kappa(V)$ over all V such that $\mathcal{A} = V\mathbf{A}V^{-1}$ is a Jordan decomposition of $\mathcal{A}$. Then, $\kappa_A \leq n \cdot e^{\frac{\|\mathcal{A}\|^2}{2\Delta^2}}$.*

We can thus conclude:

Theorem 8.5.6. *There exists a probabilistic algorithm that gets as input:*

- *An $n \times n$ irreducible, stochastic matrix $\mathcal{S}$ and a real parameter $\Delta > 0$ so that it is guaranteed that all the eigenvalues of $\mathcal{S}$ are Δ-separated (that is, $|\lambda_i - \lambda_j| \geq \Delta$ for every distinct $\lambda_i, \lambda_j \in \mathsf{Spec}(\mathcal{S})$).*
- *A parameter γ such that $\gamma'(\mathcal{S}) \geq \gamma$.*
- *An approximation $\tilde{\pi}$ of the stationary distribution π of $\mathcal{S}$, where $\|\tilde{\pi} - \pi\| \leq \tau$ and $\tau \leq \frac{\varepsilon}{(T+1)\sqrt{n}}$ for $T = \frac{8n^2}{\gamma^2}\left(1 + \log\frac{n\kappa}{\varepsilon\gamma}\right)$.*

Let $\mathcal{L}$ denote the Laplacian $\mathcal{L} = \mathcal{I} - \mathcal{S}$. The algorithm outputs a $(3\varepsilon, \delta)$-approximation of $\mathcal{L}^\star$ using

$$O\left(\log\frac{n}{\Delta\gamma\varepsilon} + \log\log\frac{1}{\delta}\right)$$

space.

8.5.4 Eulerian directed graphs

Eulerian graphs are directed graphs where the in-degree and out-degree of each vertex are the same, and so they generalize both regular directed graphs, and general undirected graphs. The stationary distribution is fully explicit (as in undirected graphs that we mentioned before). In this section we note that for Eulerian graphs γ' is always non-negligible.

Claim 8.5.7. *Let $\mathcal{S}$ be a transition matrix of a strongly connected Eulerian directed graph with m edges. Then, $\gamma'(\mathcal{S}) \geq \frac{4}{m^2}$.*

Proof. Chung [Chu05] proved that $\gamma'(\mathcal{S})$ is at least the second smallest eigenvalue μ_{n-1} (the smallest eigenvalue is 0) of

$$\mathcal{L}_G^{\mathsf{C}} = I - \frac{\Pi^{1/2}\mathcal{S}\Pi^{-1/2} + \Pi^{-1/2}\mathcal{S}^\dagger\Pi^{1/2}}{2},$$

where Π is a diagonal matrix with the stationary distribution π on the diagonal. Also, in the same paper it is proven that $\mu_{n-1} \geq \frac{4}{m^2}$, which completes the proof. □

8.5.5 Rapidly-mixing graphs

Finally, one way to approximate the stationary distribution is by taking a random walk on G until it converges. This follows directly from Lemma 8.2.3 and the fact that $\lim_{k\to\infty} P_G^k = |\mathbf{1}\rangle\langle\pi|$ (see Theorem 8.2.4). For undirected graphs (and also Eulerian directed graphs) the walk converges in polynomial time, hence, we can approximate the stationary distribution in logarithmic space, except that there is no need to do that because we have an explicit formula for the stationary distribution anyway.

For general directed graphs (even with bounded degree) the convergence rate can be exponentially small and the approach does not work. Nevertheless, there is a whole class of directed graphs, called *rapidly-mixing graphs*, that converge rapidly even though, usually, there is no explicit formula for the stationary distribution. Clearly, for graphs where the walk converges in polynomial time we can *approximate* the stationary distribution π in logarithmic space.

Chapter 9

On Derandomizing Space-Bounded Approximate-Counting Problems

As we previously mentioned, Ta-Shma [TS13] showed that the singular value decomposition and matrix inversion can be approximated in quantum log-space for well-conditioned matrices. This can be interpreted as a fully logarithmic quantum approximation scheme for both problems. We show that if $\mathsf{prBQL} = \mathsf{prBPL}$ then every fully logarithmic quantum approximation scheme can be replaced by a probabilistic one. Hence, if classical algorithms cannot approximate the above functions in logarithmic space, then there is a gap already for languages, namely, $\mathsf{prBQL} \neq \mathsf{prBPL}$.

On the way we simplify a proof of Goldreich for a similar statement for time bounded probabilistic algorithms. We show that our simplified algorithm works also in the space bounded setting (for a large set of functions) whereas Goldreich's approach does not seem to apply in the space bounded setting.

9.1 Introduction

Two well known approximation problems are approximating the singular value decomposition (SVD) of a matrix and approximating the matrix inverse. Both problems were extensively studied in the *time bounded* model, e.g., in [HMT11, ST14b, HHL09]. In the *space bounded* model it was recently shown that SVD and matrix inversion can be additively approximated by a *quantum* algorithm using only logarithmic space [TS13]. This can be interpreted as a fully logarithmic quantum approximation scheme (with additive accuracy) for the problems.

Ta-Shma's result [TS13] shows a possible gap between quantum and classical algorithms for the task of *approximating a function*. It is not clear, however, whether it also implies a possible gap between the power of quantum and classical log-space algorithms for *decision problems*. Thus, a natural question is the following. Suppose no classical log-space algorithm can approximate the SVD. Does that imply that $\mathsf{prBQL} \neq \mathsf{prBPL}$? In the contra-positive we ask whether a "de-quantumization" of decision classes (i.e., $\mathsf{prBQL} = \mathsf{prBPL}$) implies a de-quantumization of approximation schemes.

The question was also asked in the model of classical time bounded computations. A classical result by Stockmeyer [Sto85] implies that the problem of *approximate counting* of a

general NP predicate can be solved in $\mathsf{FBPP}^{\mathsf{NP}}$. Shaltiel and Umans [SU06] derandomized the result under the assumption that $\mathsf{E}_{\|}^{\mathsf{NP}}$ requires exponential size single-valued nondeterministic circuits.[1]

While in general we do not know how to approximate functions in #P better than in $\mathsf{FBPP}^{\mathsf{NP}}$, there exist #P-complete functions for which there exists a fully polynomial randomized approximation scheme (FPRAS) that *does not* require an oracle access to an NP language. Jerrum, Sinclair and Vigoda [JSV04] proved this for the permanent. Goldreich [Gol11a] showed that derandomizing the FPRAS for the permanent (or any other FPRAS) can be done under the assumption that $\mathsf{prBPP} = \mathsf{P}$:

Theorem 9.1.1 ([Gol11a])**.** *If* $\mathsf{prBPP} = \mathsf{P}$ *then every function that has an FPRAS also has such a deterministic scheme. Furthermore, for every polynomial* p*, there exists a polynomial* p' *such that if the probabilistic scheme runs in time* p*, then the deterministic one runs in time* p'*.*

In [Gol11a] Goldreich is after a "uniform" pseudorandom generator (see [Gol11a] for exact details) and Theorem 9.1.1 is a corollary of a more general technique used in his construction of a uniform PRG. Roughly speaking, Goldreich's argument works as follows: Given a probabilistic Turing machine $M(x, y)$ that approximates $f(x)$ well (where x is the input and y is the internal random coins used by M), we construct a specific sequence y_0 such that $M(x, y_0)$ also approximates $f(x)$ well. The bits of y are fixed bit by bit, where each bit is determined by a single query to a certain prBPP problem that depends on the random bits that were set so far.

In our case we deal with space bounded problems. It seems Goldreich's approach does not generalize to the space bounded case, as the random coins string is kept on the work tape, and its length is bounded by the time complexity of the Turing machine – which can be polynomial.

In this note we give an alternative (and simpler) proof that directly computes the approximated value using prBPP oracle calls, and we show this approach does generalize to a large class of functions in the space bounded model. In particular we show that if no probabilistic algorithm can approximately compute the SVD of certain well formed matrices[2] – a task that quantum algorithms can do with logarithmic space – then there is already a separation of the decision classes and $\mathsf{prBQL} \neq \mathsf{prBPL}$. We give an intuitive explanation of our proof technique at the beginning of Section 9.3 followed by a formal argument.

For a complexity decision class $\mathcal{C}$, we denote $\mathsf{pr}\mathcal{C}$ as the corresponding promise class and $\mathsf{F}\mathcal{C}$ as the corresponding function class. In Section 9.2 we define space bounded approximation schemes. We remark that we are not aware of any previous definition of space bounded approximation schemes. In Section 9.3 we present and prove our result.

[1] $\mathsf{E}_{\|}^{\mathsf{NP}}$ is the class E with *non-adaptive* oracle queries to NP. A *single-valued* nondeterministic circuit is the nonuniform analogue of $\mathsf{NP} \cap \mathsf{coNP}$.

[2] Specifically, bounded norm matrices with well-separated singular values. An $n \times n$ matrix has well-separated singular values if there exists some constant c such that for all $i \neq j$, $|\sigma_i - \sigma_j| \geq n^{-c}$.

9.2 Preliminaries

We first define approximation for real valued functions $f\colon \{0,1\}^\star \to \mathbb{R}$. We have two notions of approximation: additive and multiplicative. We say y additively approximates $f(x)$ with accuracy δ if $y \in [f(x) \pm \delta]$. We say y multiplicatively approximates $f(x)$ with accuracy δ if $y \in [(1 \pm \delta)f(x)]$.

We assume (as is customary in previous works) that the approximation algorithm M outputs a dyadic rational number, i.e., we interpret the string $M(x)$ as the rational number whose binary representation is the binary string $M(x)$. This, in particular, implies that if M has time complexity $t(n)$ then M can only output integers whose absolute value is at most $2^{t(n)}$, and this assumption is implicitly used in previous works. This assumption also implies that if $M(x)$ is a non-zero rational number then $|M(x) - 0| \geq 2^{-t(n)}$.

We say a function f is $R(n)$-bounded if $|f(x)| \leq R(n)$ for every $x \in \{0,1\}^n$ and some known uniform function R. Since we are only interested in functions that can be approximated by polynomial time algorithms we can assume without loss of generality that $R = 2^{\mathsf{poly}(n)}$, because no polynomial time algorithm can approximate a higher value. However, for specific functions f sometimes a better bound can be taken.

We begin with the time bounded model where it is customary to use *multiplicative* approximation:

Definition 9.2.1. *A fully polynomial randomized approximation scheme (FPRAS) for an $R(n)$-bounded function $f\colon \{0,1\}^n \to [0, R(n)]$ is a randomized Turing machine M that on input x, an accuracy parameter δ and a confidence parameter ε, runs in*

$$\mathsf{poly}(|x|, \delta^{-1}, \log \varepsilon^{-1}, \log R(n))$$

time and outputs $M(x) \in [(1 \pm \delta)f(x)]$ with probability at least $1 - \varepsilon$, where we interpret the string $M(x)$ as the dyadic rational number whose binary representation is given by the binary string $M(x)$. ♢

A fully polynomial (deterministic) approximation scheme (FPAS) is obtained if M in the above definition is deterministic, ε is set to 0, and the dependence on ε is removed.

In this chapter we define log-space approximation schemes, but use *additive approximation* rather than multiplicative approximation, mainly because the major examples we have for such approximation schemes only achieve additive accuracy. We define:

Definition 9.2.2. *A fully logarithmic randomized (resp. quantum) approximation scheme for an $R(n)$-bounded function $f\colon \{0,1\}^n \to [-R(n), R(n)]$ is a randomized (resp. quantum) Turing machine M that on input x, an error parameter δ and a confidence parameter ε, runs in $\mathsf{poly}(|x|, \delta^{-1}, \log \varepsilon^{-1}, \log R(n))$ time, uses $O(\log |x| + \log \delta^{-1} + \log \log \varepsilon^{-1} + \log \log R(n))$ space and outputs $M(x) \in [f(x) \pm \delta]$ with probability at least $1 - \varepsilon$.* ♢

The quantum space model we use has classical control and allows intermediate measurements, see [vMW12, TS13]. A fully logarithmic (deterministic) approximation scheme is obtained if M in the above definition is deterministic, ε is set to 0, and the dependence on ε is removed. We let FLAS abbreviate "fully logarithmic approximation scheme" and FLRAS (resp. FLQAS) the randomized (resp. quantum) versions.

If a function f computes a matrix, we say a Turing machine M approximates f if it approximates each entry of the matrix. With additive approximation this is equivalent to

approximating the matrix in the norm $\|A\|_{\max} = \max_{i,j} |A_{i,j}|$. Notice that if $\tilde{A}$ approximates an $n \times n$ matrix A in the above norm for arbitrary polynomially small δ, then it also approximates A well in the spectral norm because $\|A - \tilde{A}\|_2 \leq n^2 \|A - \tilde{A}\|_{\max} \leq n^2\delta$.

Under this notation the result in [TS13] shows that the function computing the SVD of a real matrix A that has a polynomially bounded norm and well-separated singular values and the function inverting a well-conditioned matrix with a non-negligible norm both have an FLQAS. We remark that the function computing the SVD of a matrix $A = UDV$ with polynomially bounded norm is polynomially bounded. This is because every singular value $D[i,i] = \sigma_i$ satisfies $\sigma_i \leq \|A\|_2$ and the entries of the unitary matrices U, V are obviously bounded. The same holds for the function computing the inverse of a well-conditioned matrix with a non-negligible norm, as the entries of A^{-1} are bounded by $\|A^{-1}\|_2 = \kappa(A)/\|A\|_2$ (where $\kappa(A)$ is the condition number of A).

9.3 Randomized and Quantum Space-Bounded Approximation Schemes

We first reprove Goldreich's result. We start with an intuitive explanation.

Suppose $\mathsf{prBPP} = \mathsf{P}$ and f has an FPRAS. By definition, there exists a polynomial time probabilistic algorithm M that on input x and accuracy parameter δ outputs a value $M(x,\delta)$ that with a good probability is multiplicatively δ-close to $f(x)$. Define a promise problem Π such that $\langle x, \zeta, y\rangle$ is a yes instance if $f(x) > (1+\zeta)y$ and a no instance if $f(x) \leq (1-\zeta)y$. We see that $\Pi \in \mathsf{prBPP}$ (because M essentially solves it) and by our assumption that $\mathsf{prBPP} = \mathsf{P}$ we have $\Pi \in \mathsf{P}$.

Having that we can employ a binary search to approximate $f(x)$. We start with the a-priori known lower and upper bounds $0 \leq f(x) \leq R(n) = 2^{\mathsf{poly}(n)}$ that hold for the output of any polynomial time algorithm. We then determine whether we are in the lower or upper half of this interval by calling Π. More precisely, if we know $f(x)$ belongs to the interval $[l,h]$, then with one call to the promise problem we can reduce the interval to a new one of length roughly $(\frac{1}{2}+\delta)(h-l)$. Repeating the binary search $O(\log R(n)/\delta)$ times we deterministically approximate $f(x)$ with δ multiplicative accuracy.

We now give the formal details:

Theorem 9.3.1. *Suppose* $\mathsf{prBPP} = \mathsf{P}$. *Then every function f that has an FPRAS also has an FPAS.*

Proof. Assume f has an FPRAS. Let M be a polynomial time probabilistic algorithm that on input $x \in \{0,1\}^\star$ and an accuracy parameter δ given in unary with $\frac{1}{\delta}$ bits, outputs a value $M(x,\delta)$ that with probability at least $2/3$ is multiplicatively δ-close to $f(x)$. We show how to construct a deterministic algorithm M' that on input $x \in \{0,1\}^\star$ and an accuracy parameter δ given in unary with $\frac{1}{\delta}$ bits, outputs a value $M'(x,\delta)$ that is multiplicatively δ-close to $f(x)$.

We construct the *output* $M'(x,\zeta)$ bit by bit, where each bit is determined by a single call to a prBPP promise problem. Consider the following promise problem Π:

Yes instance: $\langle x, \zeta, y\rangle$ such that $f(x) > (1+\zeta)y$.

No instance: $\langle x, \zeta, y\rangle$ such that $f(x) \leq (1-\zeta)y$.

Π is in prBPP, by the following algorithm: Run M with accuracy $\zeta' = \zeta/2$ and accept if and only if $M(x, \zeta') \geq y$. For every $\langle x, \zeta, y\rangle \in \Pi_{\text{Yes}}$, with probability at least 2/3,

$$M(x, \zeta') \geq (1-\zeta')f(x) > (1-\zeta')(1+\zeta)y \geq y.$$

Similarly, for every $\langle x, \zeta, y\rangle \in \Pi_{\text{No}}$, $M(x, \zeta') < y$ with probability at least 2/3. As we assume $\mathsf{prBPP} = \mathsf{P}$, there exists a deterministic algorithm for Π that we denote M_Π.

Let T denote the running time of M on the input $\langle x, \delta\rangle$. Set $l_0 = 0$, $h_0 = 2^T$ and $z_0 = \frac{l_0+h_0}{2}$. Notice that $l_0 \leq f(x) \leq h_0$. We now run the following binary search for $f(x)$:

For $i = 0, \ldots, \infty$ do:

- If $h_i < 2^{-T}$ output 0 and halt.
- If $\frac{h_i - l_i}{h_i + l_i} \leq \frac{\delta}{2}$ output z_i and halt.
- Query M_Π on $\langle x, \delta/4, z_i\rangle$. If the query is answered positively, set $l_{i+1} = (1-\delta/4)z_i$ and $h_{i+1} = h_i$. Otherwise, set $h_{i+1} = (1+\delta/4)z_i$ and $l_{i+1} = l_i$.
- Set $z_{i+1} = (l_{i+1} + h_{i+1})/2$.

First notice that because M_Π deterministically solves Π, we always preserve the invariance $l_i \leq f(x) \leq h_i$.

We now claim that if we output a value, then this value is a good multiplicative approximation to $f(x)$. To see that notice that we halt if either $h < 2^{-T}$ or $\frac{h-l}{h+l} \leq \frac{\delta}{2}$. In the first case $f(x) \leq h < 2^{-T}$. However, if $f(x)$ is non-zero then $f(x) \geq 2^{-T}$ (see Section 9.2). Hence $f(x) = 0$ and we output the correct value. If the latter condition happens then $|z - f(x)| \leq \delta f(x)$. To see that notice that $f(x) - z \leq h - z = h - \frac{l+h}{2} = \frac{h-l}{2} = \frac{h-l}{h+l}z \leq \frac{\delta}{2}z$. Similarly, $f(x) - z \geq -\frac{\delta}{2}z$. Thus $|f(x) - z| \leq \frac{\delta}{2}z$ which implies $z \in [(1 \pm \delta)f(x)]$.

To show that we always halt and to bound the number of iterations, let $d_i = h_i - l_i$. We claim that $d_{i+1} \leq \frac{3}{4}d_i$. This is true as $d_{i+1} \leq \frac{d_i}{2} + \frac{\delta}{4}z_i$ (immediately from the way the procedure works) and $\frac{d_i}{2} + \frac{\delta}{4}z_i \leq \frac{3}{4}d_i$ (which is true whenever $\frac{h_i - l_i}{h_i + l_i} > \frac{\delta}{2}$). Also, $\frac{h_i - l_i}{h_i + l_i} \leq 2^T d_i$ (because $h_i \geq 2^{-T}$). Hence, $\frac{h_i - l_i}{h_i + l_i} \leq 2^{2T}(\frac{3}{4})^i$ and we must stop within $O(T + \log(1/\delta))$ steps.

Altogether, the running time is $O(T + \log(1/\delta))$ times the time complexity of M_Π, which is polynomial in T as required. □

A similar procedure works for additive error by appropriately changing the promise problem Π to work with additive accuracy. We give a formal proof of this in Theorem 9.3.2 for the space bounded model, but a similar argument also works for the time bounded model.

Next, we claim that the same proof works also in the space bounded setting as long as the function f is polynomially bounded, i.e., $f : \{0,1\}^n \to [-R(n), R(n)]$ for $R(n) = \mathsf{poly}(n)$. The reason is simple: the space complexity of the deterministic machine M' constructed in the proof of Theorem 9.3.1 is $O(\log(\frac{nR(n)}{\delta}))$. Formally,

Theorem 9.3.2. *Suppose* $\mathsf{prBPL} = \mathsf{L}$. *Then every* $R = \mathsf{poly}(n)$*-bounded function* f *that has an FLRAS has an FLAS as well.*

Proof. The proof resembles the proof of Theorem 9.3.1. However, as the approximation is now additive, the promise problem Π is given by:

Yes instance: $\langle x, \zeta, y \rangle$ such that $f(x) \geq y + \zeta$.

No instance: $\langle x, \zeta, y \rangle$ such that $f(x) \leq y - \zeta$.

Π is in prBPL, by exactly the same algorithm. As $\mathsf{prBPL} = \mathsf{L}$, let M_Π be the deterministic algorithm for Π. We then run a similar loop, starting with $l \leftarrow -R(n)$ and $h \leftarrow R(n)$ and iterating as long as $|h - l| > 2\delta$ while the update for l and h is done additively.

The correctness follows from very similar reasonings as $d_{i+1} \leq \frac{d_i}{2} + \frac{\delta}{2} \leq \frac{3}{4} d_i$ in every iteration, hence the loop terminates after $\Theta(\log \delta^{-1} + \log R(n))$ iterations.

The time complexity of the FLAS is polynomial in $R(n)$ and δ^{-1}, as desired. The space required to store l, h and z, in addition to the space required to simulate M_Π is bounded by $O(\log n + \log \delta^{-1} + \log R(n))$. As $\log R(n) = O(\log n)$, we also obtain the required space constraint for an FLAS. □

The proof of Theorem 9.3.2 also generalizes to the quantum case, namely,

Theorem 9.3.3. *Suppose* $\mathsf{prBQL} = \mathsf{prBPL}$. *Then every* $R = \mathsf{poly}(n)$*-bounded function* f *that has an FLQAS has an FLRAS as well.*

Proof. Let Π be the same promise problem defined in the proof of Theorem 9.3.2. Then, by the same reasoning as in the proof of Theorem 9.3.2, $\Pi \in \mathsf{prBQL}$. Thus, if $\mathsf{prBQL} = \mathsf{prBPL}$ then $\Pi \in \mathsf{prBPL}$. We can amplify the success probability of the probabilistic algorithm so that it succeeds with probability at least $1 - \xi$ for ξ that we choose later.

Run the same binary search algorithm with the randomized algorithm for Π, and assume it uses T iterations. A similar argument to the one in the proof of Theorem 9.3.2 shows that with probability $1 - T\xi$ the algorithm outputs x to within an additive factor δ. By setting $\xi = \frac{\varepsilon}{T}$, the FLRAS with accuracy parameter δ and confidence parameter ε runs in time $T \cdot \mathsf{poly}(n, \delta^{-1}, \log \xi^{-1}, \log R(n)) = \mathsf{poly}(n, \delta^{-1}, \log \varepsilon^{-1})$ and uses $O(\log R(n) + \log n + \log \delta^{-1} + \log \log \xi^{-1}) = O(\log n + \log \delta^{-1} + \log \log \varepsilon^{-1} + \log \log R(n))$ space, as desired. □

As explained before, [TS13] shows that the function computing the SVD of a real matrix A that has a polynomially bounded norm and well-separated singular values, and the function inverting a well-conditioned matrix with a non-negligible norm both have an FLQAS, and both compute a polynomially bounded function. Thus, either $\mathsf{prBQL} \neq \mathsf{prBPL}$, or else there must exist an FLRAS for both problems, implying that approximately inverting a matrix is essentially in prBPL.

Conclusion

In Chapter 3, we showed that explicit two-source extractors that support small entropies can be constructed via non-malleable extractors that support small entropies and have a short seed. A bit more formally, an explicit ε-error non-malleable extractor for sources of length n that supports min-entropy $f(n, \varepsilon)$ and has seed-length $O\left(f(n, \varepsilon)\right)$ would imply an explicit, constant-error, two-source extractor for n-bit sources supporting min-entropy $O(f(n, \frac{1}{\mathsf{poly}(n)}))$. Nowadays, better and better non-malleable extractors are being constructed, and the hope is that the above scheme will indeed be beneficial in achieving an explicit two-source extractor supporting $O(\log n)$ min-entropy.

All constructions of non-malleable extractors that work for small min-entropies use *alternating extraction* protocols, and the recent ones use primitives such as *advice generators*, *correlation breakers with advice* and *independence-preserving mergers*. Prolific as they are, such methods of breaking correlations between cleverly generated random variables have their drawbacks. For one, they are relatively involved. Also, current constructions of independence-preserving mergers (which are the crux of good non-malleable extractors) are sub-optimal, and coming up with new observations will be crucial in order to avoid losses incurred by the alternating extractions. Alternatively, other than improving independent-preserving mergers, coming up with a new and elegant construction of non-malleable extractors (even with worse parameters) is a dignified goal and would certainly improve our understanding of these objects.

Constructing new non-malleable extractors can also be beneficial for constructing *low error* two-source extractors, as we exemplified in Chapter 4. The seed-length's dependence on the non-malleability parameter t is nowadays far from being optimal, and all recent constructions have a multiplicative dependence on t^2 rather than an additive dependence. Refining the current (rather crude) analysis of non-malleable extractors with respect to t would be rather exciting, and again, coming up with an alternative construction would be even more so. We also note that our low-error condenser of Chapter 5 might be used as a stepping-stone for constructing low-error two-source extractors, as often has happened in the past.

In Chapter 6, we demonstrated how new techniques in extractor theory can be used for constructing binary erasure list-decodable codes and strong dispersers outputting one bit. As already mentioned, constructing good *linear* strong dispersers, that possibly output many bits, is an interesting challenge. Such dispersers, I believe, can also be used to construct other pseudorandom objects, most notably good affine extractors.

In the area of space-bounded probabilistic computation, we introduced, in [DSTS17], a new BPL-complete problem, namely, approximating the spectral gap of a stochastic operator

having a real second eigenvalue. Analyzing operators that arise from *directed* graphs was also done in Chapter 8, where we approximated the solution of a Laplacian system in probabilistic logspace, even for Laplacian of certain directed graphs. Studying directed graphs is hard, but bears viable approaches for derandomizing small-space computation. For example, it was shown in [RTV06] that a pseudorandom walk generator for regular directed graphs is already sufficient to derandomize RL.

From an algorithmic point of view, it would be interesting to find fully logarithmic probabilistic/quantum approximation schemes for problems whose *exact* variants lie in DET. A complexity-theoretic incentive was given in Chapter 9, but nonetheless any novel algorithm would certainly improve our theoretical insight.

Bibliography

[Abb72] H.L. Abbott. Lower bounds for some Ramsey numbers. *Discrete Mathematics*, 2(4):289–293, 1972.

[AGHP92] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple Construction of Almost k-wise Independent Random Variables. *Random Struct. Algorithms*, 3(3):289–304, 1992. Preliminary version in the *31st Annual IEEE Symposium on Foundations of Computer Science (FOCS 1990)*.

[AGM03] Noga Alon, Oded Goldreich, and Yishay Mansour. Almost k-wise independence versus k-wise independence. *Information Processing Letters*, 88(3):107–110, 2003.

[AKL+79] Romas Aleliunas, Richard M Karp, Richard J Lipton, Laszlo Lovasz, and Charles Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *Proceedings of the 20th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1979)*, pages 218–223. IEEE, 1979.

[AL93] Miklós Ajtai and Nathan Linial. The influence of large coalitions. *Combinatorica*, 13(2):129–145, 1993.

[Alo98] Noga Alon. The Shannon capacity of a union. *Combinatorica*, 18(3):301–310, 1998.

[ASWZ96] Roy Armoni, Michael Saks, Avi Wigderson, and Shiyu Zhou. Discrepancy sets and pseudorandom generators for combinatorial rectangles. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1996)*, pages 412–421. IEEE, 1996.

[BACD+18] Avraham Ben-Aroya, Eshan Chattopadhyay, Dean Doron, Xin Li, and Amnon Ta-Shma. A new approach for constructing low-error, two-source extractors. *LIPIcs-Leibniz International Proceedings in Informatics. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik*, 2018.

[BACDTS18] Avraham Ben-Aroya, Gil Cohen, Dean Doron, and Amnon Ta-Shma. Two-source condensers with low error and small entropy gap via entropy-resilient functions. In *Electronic Colloquium on Computational Complexity (ECCC)*, 2018.

[BADTS17] Avraham Ben-Aroya, Dean Doron, and Amnon Ta-Shma. An efficient reduction from two-source to non-malleable extractors: achieving near-logarithmic min-entropy. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC 2017)*, pages 1185–1194. ACM, 2017.

[BADTS18] Avraham Ben-Aroya, Dean Doron, and Amnon Ta-Shma. Near-optimal strong dispersers, erasure list-decodable codes and friends. In *Electronic Colloquium on Computational Complexity (ECCC)*, 2018.

[Bar06] Boaz Barak. A simple explicit construction of an $n^{\tilde{O}(\log n)}$-Ramsey graph. *arXiv preprint math/0601651*, 2006.

[Bau12] Frank Bauer. Normalized graph Laplacians for directed graphs. *Linear Algebra and its Applications*, 436(11):4193–4222, 2012.

[BBR88] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

[BCG18] Mark Braverman, Gil Cohen, and Sumegha Garg. Hitting sets with near-optimal error for read-once branching programs. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC 2018)*, pages 353–362. ACM, 2018.

[BDVY13] Andrej Bogdanov, Zeev Dvir, Elad Verbin, and Amir Yehudayoff. Pseudorandomness for width-2 branching programs. *Theory of Computing*, 9:283–293, 2013.

[BGK06] Jean Bourgain, A. A. Glibichuk, and Sergei Vladimirovich Konyagin. Estimates for the number of sums and products and for exponential sums in fields of prime order. *Journal of the London Mathematical Society*, 73(2):380–398, 2006.

[BIG03] Adi Ben-Israel and Thomas N.E. Greville. *Generalized Inverses: Theory and Applications*, volume 15. Springer, 2003.

[BIW06] Boaz Barak, Russell Impagliazzo, and Avi Wigderson. Extracting randomness using few independent sources. *SIAM Journal on Computing*, 36(4):1095–1118, 2006.

[BKS+10] Boaz Barak, Guy Kindler, Ronen Shaltiel, Benny Sudakov, and Avi Wigderson. Simulating independence: New constructions of condensers, ramsey graphs, dispersers, and extractors. *Journal of the ACM*, 57(4):20, 2010.

[BKT04] Jean Bourgain, Nets Katz, and Terence Tao. A sum-product estimate in finite fields, and applications. *Geometric & Functional Analysis GAFA*, 14(1):27–57, 2004.

[BL92] Anders Björner and László Lovász. Chip-firing games on directed graphs. *Journal of Algebraic Combinatorics*, 1(4):305–328, 1992.

[BMRV02] Harry Buhrman, Peter Bro Miltersen, Jaikumar Radhakrishnan, and Srinivasan Venkatesh. Are bitvectors optimal? *SIAM Journal on Computing*, 31(6):1723–1744, 2002.

[BOL85] Michael Ben-Or and Nathan Linial. Collective coin flipping, robust voting schemes and minima of Banzhaf values. In *Proceedings of the 26th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 408–416. IEEE, 1985.

[Bou05] Jean Bourgain. More on the sum-product phenomenon in prime fields and its applications. *International Journal of Number Theory*, 1(01):1–32, 2005.

[Bou07] Jean Bourgain. On the construction of affine extractors. *GAFA Geometric And Functional Analysis*, 17(1):33–57, 2007.

[Bra10] Mark Braverman. Polylogarithmic independence fools AC0 circuits. *Journal of the ACM*, 57(5):28, 2010.

[Bro88] William Clough Brown. *A Second Course in Linear Algebra*. Wiley-Interscience, 1988.

[Bro91] Richard Bronson. *Matrix Methods: an Introduction*. Gulf Professional Publishing, 1991.

[BRRY10] Mark Braverman, Anup Rao, Ran Raz, and Amir Yehudayoff. Pseudorandom generators for regular branching programs. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010)*, pages 40–47. IEEE, 2010.

[BRRY14] Mark Braverman, Anup Rao, Ran Raz, and Amir Yehudayoff. Pseudorandom Generators for Regular Branching Programs. *SIAM J. Comput.*, 43(3):973–986, 2014.

[BRSW12] Boaz Barak, Anup Rao, Ronen Shaltiel, and Avi Wigderson. 2-source dispersers for $n^{o(1)}$ entropy, and Ramsey graphs beating the Frankl-Wilson construction. *Annals of Mathematics*, pages 1483–1543, 2012.

[BSRZ15] Eli Ben-Sasson and Noga Ron-Zewi. From affine to two-source extractors via approximate duality. *SIAM Journal on Computing*, 44(6):1670–1697, 2015.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[CGH+85] Benny Chor, Oded Goldreich, Johan Håstad, Joel Freidmann, Steven Rudich, and Roman Smolensky. The bit extraction problem or t-resilient functions. In *Proceedings of the 26th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 396–407. IEEE, 1985.

[CGL16] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC 2016)*, pages 285–298. ACM, 2016.

[CHHL18] Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, and Shachar Lovett. Pseudorandom generators from polarizing random walks. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 102. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[Chu81] Fan R.K. Chung. A note on constructive methods for Ramsey numbers. *Journal of Graph Theory*, 5(1):109–113, 1981.

[Chu96] Fan R. K. Chung. Laplacian of graphs and Cheeger's inequalities. *Combinatorics, Paul Erdos is Eighty*, 2(157-172):13–2, 1996.

[Chu97] Fan R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[Chu05] Fan R. K. Chung. Laplacians and the Cheeger inequality for directed graphs. *Annals of Combinatorics*, 9(1):1–19, 2005.

[CI17] Mahdi Cheraghchi and Piotr Indyk. Nearly optimal deterministic algorithm for sparse walsh-hadamard transform. *ACM Transactions on Algorithms (TALG)*, 13(3):34, 2017.

[CKP+16] Michael B. Cohen, Jonathan Kelner, John Peebles, Richard Peng, Aaron Sidford, and Adrian Vladu. Faster algorithms for computing the stationary distribution, simulating random walks, and more. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2016)*, pages 583–592. IEEE, 2016.

[CL16] Eshan Chattopadhyay and Xin Li. Explicit non-malleable extractors, multi-source extractors, and almost optimal privacy amplification protocols. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2016)*, pages 158–167. IEEE, 2016.

[Coh16a] Gil Cohen. Local correlation breakers and applications to three-source extractors and mergers. *SIAM Journal on Computing*, 45(4):1297–1338, 2016.

[Coh16b] Gil Cohen. Making the most of advice: new correlation breakers and their applications. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2016)*, pages 188–196. IEEE, 2016.

[Coh16c] Gil Cohen. Non-malleable extractors – new tools and improved constructions. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 50. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[Coh16d] Gil Cohen. Non-malleable extractors with logarithmic seeds. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 23, page 30, 2016.

[Coh16e] Gil Cohen. Two-source dispersers for polylogarithmic entropy and improved Ramsey graphs. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC 2016)*, pages 278–284. ACM, 2016.

[Coh16f] Gil Cohen. Two-source extractors for quasi-logarithmic min-entropy and improved privacy amplification protocols. In *Electronic Colloquium on Computational Complexity (ECCC)*, 2016.

[Coh17] Gil Cohen. Towards optimal two-source extractors and Ramsey graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1157–1170. ACM, 2017.

[Coo85] Stephen A. Cook. A taxonomy of problems with fast parallel algorithms. *Information and control*, 64(1-3):2–22, 1985.

[CRS14] Gil Cohen, Ran Raz, and Gil Segev. Nonmalleable extractors with short seeds and applications to privacy amplification. *SIAM Journal on Computing*, 43(2):450–476, 2014.

[CS15] Gil Cohen and Igor Shinkar. Zero-fixing extractors for sub-logarithmic entropy. In *International Colloquium on Automata, Languages, and Programming*, pages 343–354. Springer, 2015.

[CS16] Gil Cohen and Leonard J. Schulman. Extractors for near logarithmic min-entropy. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2016)*, pages 178–187. IEEE, 2016.

[CS17] Gil Cohen and Igor Shinkar. Personal communication, 2017.

[Csa76] László Csanky. Fast parallel matrix inversion algorithms. *SIAM Journal on Computing*, 5(4):618–623, 1976.

[CW89] Aviad Cohen and Avi Wigderson. Dispersers, deterministic amplification, and weak random sources. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1989)*, pages 14–19. IEEE, 1989.

[CZ16] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC 2016)*, pages 670–683. ACM, 2016.

[De11] Anindya De. Pseudorandomness for permutation and regular branching programs. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity (CCC 2011)*, pages 221–231, 2011.

[Dem97] James W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.

[DGJ+10] Ilias Diakonikolas, Parikshit Gopalan, Ragesh Jaiswal, Rocco A. Servedio, and Emanuele Viola. Bounded independence fools halfspaces. *SIAM Journal on Computing*, 39(8):3441–3462, 2010.

[DKSS13] Zeev Dvir, Swastik Kopparty, Shubhangi Saraf, and Madhu Sudan. Extensions to the method of multiplicities, with applications to Kakeya sets and mergers. *SIAM Journal on Computing*, 42(6):2305–2328, 2013.

[DLGTS17] Dean Doron, François Le Gall, and Amnon Ta-Shma. Probabilistic logarithmic-space algorithms for Laplacian solvers. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 81. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[DLWZ14] Yevgeniy Dodis, Xin Li, Trevor D. Wooley, and David Zuckerman. Privacy amplification and nonmalleable extractors via character sums. *SIAM Journal on Computing*, 43(2):800–830, 2014.

[DPW14] Yevgeniy Dodis, Krzysztof Pietrzak, and Daniel Wichs. Key derivation without entropy waste. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 93–110. Springer, 2014.

[DSTS17] Dean Doron, Amir Sarid, and Amnon Ta-Shma. On approximating the eigenvalues of stochastic matrices in probabilistic logspace. *Computational Complexity*, 26(2):393–420, 2017.

[DTS15a] Dean Doron and Amnon Ta-Shma. On the de-randomization of space-bounded approximate counting problems. *Information Processing Letters*, 115(10):750–753, 2015.

[DTS15b] Dean Doron and Amnon Ta-Shma. On the problem of approximating the eigenvalues of undirected graphs in probabilistic logspace. In *International Colloquium on Automata, Languages, and Programming*, pages 419–431. Springer, 2015.

[DW09] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC 2009)*, pages 601–610. ACM, 2009.

[DW11] Zeev Dvir and Avi Wigderson. Kakeya sets, new mergers, and old extractors. *SIAM Journal on Computing*, 40(3):778–792, 2011.

[Erd47] Paul Erdös. Some remarks on the theory of graphs. *Bulletin of the American Mathematical Society*, 53(4):292–294, 1947.

[FL18] Bill Fefferman and Cedric Yen-Yu Lin. A complete characterization of unitary quantum space. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 94. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[Fra77] Peter Frankl. A constructive lower bound for Ramsey numbers. *Ars Combinatoria*, 3(297-302):28, 1977.

[FW81] Peter Frankl and Richard M. Wilson. Intersection theorems with geometric consequences. *Combinatorica*, 1(4):357–368, 1981.

[Gab11] Ariel Gabizon. Deterministic extractors for bit-fixing sources by obtaining an independent seed. In *Deterministic Extraction from Weak Random Sources*, pages 11–32. Springer, 2011.

[GI02] Venkatesan Guruswami and Piotr Indyk. Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC 2002)*, pages 812–821. ACM, 2002.

[GKM15] Parikshit Gopalan, Daniek Kane, and Raghu Meka. Pseudorandomness via the discrete fourier transform. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2015)*, pages 903–922. IEEE, 2015.

[GKRTS05] Ronen Gradwohl, Guy Kindler, Omer Reingold, and Amnon Ta-Shma. On the error parameter of dispersers. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 294–305. Springer, 2005.

[Gly87] Peter W. Glynn. Upper bounds on Poisson tail probabilities. *Operations Research Letters*, 6(1):9–14, 1987.

[GMR+12] Parikshit Gopalan, Raghu Meka, Omer Reingold, Luca Trevisan, and Salil Vadhan. Better pseudorandom generators from milder pseudorandom restrictions. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2012)*, pages 120–129. IEEE, 2012.

[GMRZ13] Parikshit Gopalan, Raghu Meka, Omer Reingold, and David Zuckerman. Pseudorandom generators for combinatorial shapes. *SIAM Journal on Computing*, 42(3):1051–1076, 2013.

[God82] Chris Godsil. Eigenvalues of graphs and digraphs. *Linear Algebra and its Applications*, 46:43–50, 1982.

[Gol11a] Oded Goldreich. In a world of P=BPP. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 191–232. Springer, 2011.

[Gol11b] Oded Goldreich. A sample of samplers: a computational perspective on sampling. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 302–332. Springer, 2011.

[GOWZ10] Parikshit Gopalan, Ryan O'Donnell, Yi Wu, and David Zuckerman. Fooling functions of halfspaces under product distributions. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity (CCC 2010)*, pages 223–234. IEEE, 2010.

[GR13] Chris Godsil and Gordon F. Royle. *Algebraic Graph Theory*. Springer, 2013.

[Gro01] Vince Grolmusz. Low rank co-diagonal matrices and Ramsey graphs. *Journal of combinatorics*, 7(1):R15–R15, 2001.

[Gro14] Codruţ Grosu. $\mathbb{F}_p$ is locally like $\mathbb{C}$. *Journal of the London Mathematical Society*, 89(3):724–744, 2014.

[GRS06] Ariel Gabizon, Ran Raz, and Ronen Shaltiel. Deterministic extractors for bit-fixing sources by obtaining an independent seed. *SIAM Journal on Computing*, 36(4):1072–1094, 2006.

[Gur03] Venkatesan Guruswami. List decoding from erasures: Bounds and code constructions. *IEEE Transactions on Information Theory*, 49(11):2826–2833, 2003.

[Gur04a] Venkatesan Guruswami. Better extractors for better codes? In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC 2004)*, pages 436–444. ACM, 2004.

[Gur04b] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes (Winning Thesis of the 2002 ACM Doctoral Dissertation Competition)*, volume 3282 of *Lecture Notes in Computer Science*. Springer, 2004.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. *Journal of the ACM*, 56(4):20, 2009.

[GVL96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, 3rd edition, 1996.

[HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009.

[Hig08] Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, 2008.

[HMT11] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[HS16] Prahladh Harsha and Srikanth Srinivasan. On polynomial approximations to AC0. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2016.

[HZ18] William Hoza and David Zuckerman. Simple optimal hitting sets for small-success RL. In *Electronic Colloquium on Computational Complexity (ECCC)*, 2018.

[ILL89] Russell Impagliazzo, Leonid A Levin, and Michael Luby. Pseudo-random generation from one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC 1989)*, pages 12–24. ACM, 1989.

[INW94] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing (STOC 1994)*, pages 356–364. ACM, 1994.

[JSV04] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM*, 51(4):671–697, 2004.

[KKL88] Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on Boolean functions. In *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1988)*, pages 68–80. IEEE, 1988.

[KNP11] Michal Koucký, Prajakta Nimbhorkar, and Pavel Pudlák. Pseudorandom generators for group products: extended abstract. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC 2011)*, pages 263–272, 2011.

[KOSZ13] Jonathan A. Kelner, Lorenzo Orecchia, Aaron Sidford, and Zeyuan Allen Zhu. A simple, combinatorial algorithm for solving SDD systems in nearly-linear time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC 2013)*, pages 911–920. ACM, 2013.

[KVM02] Adam R. Klivans and Dieter Van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM Journal on Computing*, 31(5):1501–1526, 2002.

[KZ06] Jesse Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM Journal on Computing*, 36(5):1231–1247, 2006.

[Li11] Xin Li. Improved constructions of three source extractors. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity (CCC 2011)*, pages 126–136. IEEE, 2011.

[Li12a] Xin Li. Design extractors, non-malleable condensers and privacy amplification. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 837–854. ACM, 2012.

[Li12b] Xin Li. Non-malleable extractors, two-source extractors and privacy amplification. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2012)*, pages 688–697. IEEE, 2012.

[Li13a] Xin Li. Extractors for a constant number of independent sources with polylogarithmic min-entropy. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, pages 100–109. IEEE, 2013.

[Li13b] Xin Li. New independent source extractors with exponential improvement. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC 2013)*, pages 783–792. ACM, 2013.

[Li15a] Xin Li. Non-malleable condensers for arbitrary min-entropy, and almost optimal protocols for privacy amplification. In *Theory of Cryptography Conference*, pages 502–531. Springer, 2015.

[Li15b] Xin Li. Three-source extractors for polylogarithmic min-entropy. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2015)*, pages 863–882. IEEE, 2015.

[Li16] Xin Li. Improved two-source extractors, and affine extractors for polylogarithmic entropy. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2016)*, pages 168–177. IEEE, 2016.

[Li17] Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2017)*, pages 1144–1156. IEEE, 2017.

[Li18] Xin Li. Non-malleable extractors and non-malleable codes: partially optimal constructions. *arXiv preprint arXiv:1804.04005*, 2018.

[Lov14] Shachar Lovett. Recent advances on the log-rank conjecture in communication complexity. *Bulletin of EATCS*, 1(112), 2014.

[Lov16] Shachar Lovett. Communication is bounded by root of rank. *Journal of the ACM*, 63(1):1, 2016.

[LRTV09] Shachar Lovett, Omer Reingold, Luca Trevisan, and Salil Vadhan. Pseudorandom bit generators that fool modular sums. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 615–630. Springer, 2009.

[LRVW03] Chi-Jen Lu, Omer Reingold, Salil Vadhan, and Avi Wigderson. Extractors: optimal up to constant factors. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC 2003)*, pages 602–611. ACM, 2003.

[Mek17] Raghu Meka. Explicit resilient functions matching Ajtai-Linial. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2017)*, pages 1132–1148. SIAM, 2017.

[MRSV17] Jack Murtagh, Omer Reingold, Aaron Sidford, and Salil Vadhan. Derandomization beyond connectivity: undirected Laplacian systems in nearly logarithmic space. *arXiv preprint arXiv:1708.04634*, 2017.

[MRT18] Raghu Meka, Omer Reingold, and Avishay Tal. Pseudorandom generators for width-3 branching programs. *arXiv preprint arXiv:1806.04256*, 2018.

[MRZ14] Raghu Meka, Omer Reingold, and Yuan Zhou. Deterministic coupon collection and better strong dispersers. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 28. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[MZ13] Raghu Meka and David Zuckerman. Pseudorandom generators for polynomial threshold functions. *SIAM Journal on Computing*, 42(3):1275–1301, 2013.

[Nag75] Zs Nagy. A constructive estimation of the Ramsey numbers. *Mat. Lapok*, 23:301–302, 1975.

[Nao92] Moni Naor. Constructing Ramsey graphs from small probability spaces. *IBM Research Report RJ*, 8810, 1992.

[Neu51] J. V. Neumann. Various techniques used in connection with random digits. *Applied Math Series*, 12(1):36–38, 1951.

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[Nis94] Noam Nisan. RL $\subseteq$ SC. *Computational Complexity*, 4(1):1–11, 1994.

[NL82] Victor Neumann-Lara. The dichromatic number of a digraph. *Journal of Combinatorial Theory, Series B*, 33(3):265–270, 1982.

[NN93] Joseph Naor and Moni Naor. Small-bias probability spaces: efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, 1993.

[NTS99] Noam Nisan and Amnon Ta-Shma. Extracting randomness: a survey and new constructions. *Journal of Computer and System Sciences*, 58(1):148–173, 1999.

[NW95] Noam Nisan and Avi Wigderson. On rank vs. communication complexity. *Combinatorica*, 15(4):557–565, 1995.

[NZ93] Noam Nisan and David Zuckerman. More deterministic simulation in logspace. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC 1993)*, pages 235–244. ACM, 1993.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

[PS14] Richard Peng and Daniel A. Spielman. An efficient parallel solver for SDD linear systems. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 333–342. ACM, 2014.

[Ram30] F.P. Ramsey. On a Problem of Formal Logic. *Proceedings of the London Mathematical Society*, 2(1):264–286, 1930.

[Rao08] A. Rao. A 2-source almost-extractor for linear entropy. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 549–556. Springer, 2008.

[Rao09a] Anup Rao. Extractors for a constant number of polynomially small min-entropy independent sources. *SIAM Journal on Computing*, 39(1):168–194, 2009.

[Rao09b] Anup Rao. Extractors for low-weight affine sources. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity (CCC 2009)*, pages 95–101. IEEE, 2009.

[Raz05] Ran Raz. Extractors with weak random seeds. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC 2005)*, pages 11–20. ACM, 2005.

[Rei08] Omer Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4), 2008.

[RRV99] Ran Raz, Omer Reingold, and Salil Vadhan. Error reduction for extractors. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1999)*, pages 191–201. IEEE, 1999.

[RSW00] Omer Reingold, Ronen Shaltiel, and Avi Wigderson. Extracting randomness via repeated condensing. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2000)*, pages 22–31. IEEE, 2000.

[RTS00] Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000.

[RTV06] Omer Reingold, Luca Trevisan, and Salil Vadhan. Pseudorandom walks on regular digraphs and the RL vs. L problem. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC 2006)*, pages 457–466. ACM, 2006.

[RV05] Eyal Rozenman and Salil Vadhan. Derandomized squaring of graphs. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 436–447. Springer, 2005.

[RVW02] Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of mathematics*, pages 157–187, 2002.

[Sav70] Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of computer and system sciences*, 4(2):177–192, 1970.

[Sha02] Ronen Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77(67-95):10, 2002.

[Sha11] Ronen Shaltiel. An introduction to randomness extractors. In *International Colloquium on Automata, Languages, and Programming*, pages 21–41. Springer, 2011.

[Sip88] Michael Sipser. Expanders, randomness, or time versus space. *Journal of Computer and System Sciences*, 36(3):379–383, 1988.

[Smi67] Russell A. Smith. The condition numbers of the matrix eigenvalue problem. *Numerische Mathematik*, 10(3):232–240, 1967.

[SSZ98] Michael Saks, Aravind Srinivasan, and Shiyu Zhou. Explicit OR-dispersers with polylogarithmic degree. *Journal of the ACM*, 45(1):123–154, 1998.

[ST04] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC 2004)*, pages 81–90, 2004.

[ST11] Daniel A. Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011.

[ST13] Daniel A. Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM Journal on Computing*, 42(1):1–26, 2013.

[ST14a] Daniel A. Spielman and Shang-Hua Teng. Nearly Linear Time Algorithms for Preconditioning and Solving Symmetric, Diagonally Dominant Linear Systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014.

[ST14b] Daniel A Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014.

[Ste12] Thomas Steinke. Pseudorandomness for Permutation Branching Programs Without the Group Theory. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:83, 2012.

[Sto85] Larry Stockmeyer. On approximation algorithms for #P. *SIAM Journal on Computing*, 14(4):849–861, 1985.

[STSZ06] Shmuel Safra, Amnon Ta-Shma, and David Zuckerman. Extractors from Reed–Muller codes. *Journal of Computer and System Sciences*, 72(5):786–812, 2006.

[SU05] Ronen Shaltiel and Christopher Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *Journal of the ACM*, 52(2):172–216, 2005.

[SU06] Ronen Shaltiel and Christopher Umans. Pseudorandomness for approximate counting and sampling. *Computational Complexity*, 15(4):298–341, 2006.

[SZ99a] Michael Saks and Shiyu Zhou. $BP_HSPACE(S) \subseteq DSPACE(S^{3/2})$. *Journal of Computer and System Sciences*, 58(2):376–403, 1999.

[SZ99b] Aravind Srinivasan and David Zuckerman. Computing with very weak random sources. *SIAM Journal on Computing*, 28(4):1433–1459, 1999.

[Tal17] Avishay Tal. Tight bounds on the Fourier spectrum of AC0. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 79. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[Tre01] Luca Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, 48(4):860–879, 2001.

[TS96] Amnon Ta-Shma. On extracting randomness from weak random sources. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC 1996)*, pages 276–285. ACM, 1996.

[TS02] Amnon Ta-Shma. Almost optimal dispersers. *Combinatorica*, 22(1):123–145, 2002.

[TS13] Amnon Ta-Shma. Inverting well conditioned matrices in quantum logspace. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC 2013)*, pages 881–890. ACM, 2013.

[TSU12] Amnon Ta-Shma and Christopher Umans. Better condensers and new extractors from Parvaresh-Vardy codes. In *Proceedings of the 27th Annual IEEE Conference on Computational Complexity (CCC 2012)*, pages 309–315. IEEE, 2012.

[TSUZ07] Amnon Ta-Shma, Christopher Umans, and David Zuckerman. Lossless condensers, unbalanced expanders, and extractors. *Combinatorica*, 27(2):213–240, 2007.

[TV00] Luca Trevisan and Salil Vadhan. Extracting randomness from samplable distributions. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2000)*, pages 32–42. IEEE, 2000.

[Uma03] Christopher Umans. Pseudo-random generators for all hardnesses. *Journal of Computer and System Sciences*, 67(2):419–440, 2003.

[Vad12] Salil P. Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

[Val79] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[Vaz86] Umesh V. Vazirani. *Randomness, Adversaries and Computation*. University of California, Berkeley, 1986.

[Vio09] Emanuele Viola. The sum of d small-bias generators fools polynomials of degree d. *Computational Complexity*, 18(2):209–217, 2009.

[Vio14] Emanuele Viola. Extractors for circuit sources. *SIAM Journal on Computing*, 43(2):655–672, 2014.

[Vis13] Nisheeth K. Vishnoi. *$Lx = b$ — Laplacian Solvers and their Algorithmic Applications*. Now publishers, 2013.

[vMW12] Dieter van Melkebeek and Thomas Watson. Time-space efficient simulations of quantum computations. *Theory of Computing*, 8(1):1–51, 2012.

[Wig09] Avi Wigderson. Randomness extractors–applications and constructions. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 4. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2009.

[Yeh11] Amir Yehudayoff. Affine extractors over prime fields. *Combinatorica*, 31(2):245–256, 2011.

[Zuc90] David Zuckerman. General weak random sources. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science (FOCS 1990)*, pages 534–543. IEEE, 1990.

[Zuc96a] David Zuckerman. On unapproximable versions of NP-complete problems. *SIAM Journal on Computing*, 25(6):1293–1304, 1996.

[Zuc96b] David Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16(4-5):367–391, 1996.

[Zuc97] David Zuckerman. Randomness-optimal oblivious sampling. *Random Structures and Algorithms*, 11(4):345–367, 1997.

[Zuc07] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3:103–128, 2007.

של תא-שמע: האם ניתן לקרב את הערכים הסינגולריים כבר ב- BPL? לשאלה זו נתנו תשובה חיובית, אך רק עבור מטריצות סטוכסטיות, הרמיטיות ושגיאה קבועה. פפרמן ולין [FL18] נתנו שתי בעיות שלמות ל- BQL – קירוב ההפכי והערך-עצמי המינימלי של מטריצה חיובית-מוגדרת, עם שגיאה פולינומית-קטנה.

בפרק 8 אנו דנים בבעיה חשובה הקשורה בהיפוך מטריצות – קירוב לפתרון מערכת משוואות לינאריות הניתנות ע״י לפלסיאן של גרף. אנו נותנים אלגוריתם הסתברותי הרץ בזכרון לוגריתמי עבור גרפים לא-מכוונים ועבור גרפים מכוונים עם זמן ערבוב פולינומי. האלגוריתם משיג שגיאה פולינומית-נמוכה, כמו האלגוריתם הקוונטי. בהיבט סיבוכיות זכרון דטרמיניסטית, תוך שימוש ב- [SZ99a] עבודתנו נותנת אלגוריתם קירוב דטרמיניסטי המשתמש ב- $O(\log^{1.5} n)$ זכרון. תוצאה זו שופרה לאחר מכן ע״י מורטאג ואחרים [MRSV17], תוך שימוש בשיטות אחרות, ל- $O(\log n \log\log n)$.

אלגוריתם הקירוב שלנו למערכת המשוואות הוא בעצם אלגוריתם לקירוב של **היפוך מוכלל** של הלפלסיאן. אנו משיגים זאת ע״י טיפול בגרעין הלא-טריוויאלי של האופרטור ואז יישום טכניקות מהילוכים מקריים על גרפים מכוונים כדי לקרב את האופרטור ההפכי על תת-המרחב ההפיך.

המסר העולה מהדיון הינו שמחלקות הסיבוכיות מוגבלות הזכרון הדטרמיניסטיות, ההסתברותיות והקוונטיות יכולות להיות מאופינות ע״י בעיות **קירוב** אלגבריות, שפתרון מדויק עבורן ניתן לחשב כבר ב- DET. ההבדל בין המחלקות נעוץ בסוג האופרטורים שעבורם הבעיות מוגדרות. אם כך, המצב דומה לקשר בין BPP, בה ניתן לקרב את הפרמננט [JSV04] לבין המחלקה #P, בה ניתן לחשב את הפרמננט בדיוק [Val79]. אנו מאמינים שצורת ההסתכלות הזו אינה רק חשובה בפני עצמה (דהיינו, לטובת פיתוח אלגוריתמים מוגבלי זכרון לבעיות נפוצות) אלא גם עשויה לשפוך אור על החוזק והחולשה של המודלים השונים של חישוב מוגבל-זכרון.

בעיות ספירה מקורבת בזכרון מוגבל. ישנן מגוון בעיות שעודן פתוחות. האם זה עדיין BPL-קשה לקרב את הפער הספקטרלי אם האופרטור הוא סטוכסטי והרמיטי? או, האם ניתן לקרב ב- BPL את הערך-עצמי השני של אופרטור כללי?

באופן כללי, האם ניתן לקרב את הערכים הסינגולריים של כל אופרטור כבר ב- BPL? תשובה חיוביות לכך תגרור קירוב ב- BPL של מגוון בעיות אלגבריות שכיום רק ידוע שהן ב- DET. בפרק 9 אנו מוכיחים כי תשובה שלילית כבר תגרור הפרדה בין מחלקות ההכרעה BQL ו- BPL. בהסתכלות רחבה יותר, אנו דנים בדה-רנדומיזציה של בעיות ספירה מקורבת ומראים שבמודל המוגבל-זכרון, תוצאת אי-קירוב של בעיות אלו תגרור הפרדה בין מחלקות ההכרעה המתאימות.

התוצאה של סאקס וז׳ו לא שופרה מאז, אך עבודה רבה הושקעה בבנייתם של מחוללים עבור מחלקות חלשות יותר ב- RL. דוגמאות לכך הן מגוון מגבלות על תכניות מסתעפות עם רוחב חסום [BRRY10, De11, BDVY13, CHHL18, MRT18], נוסחאות בקריאה-יחידה, חצאי-מישור, פונקציות סף פולינומיות, מבחנים מודולריים, מלבנים וצורות קומבינטוריות וצורות פורייה – המכלילות מגוון מהמחלקות שהוזכרו לעיל [LRTV09, Vio09, DGJ+10, GOWZ10, GMR+12, GMRZ13, MZ13, GKM15]. באופן טיפוסי, אורך הגרעין של מחוללים אלו הינו קרוב לאופטימלי.

בעיה שלמה ל- BPL. פרט לגישות מתחום הפסאודו-אקראיות, המנסות לבנות מחולל פסאודו-אקראיות עבור BPL, ניתן לנסות לבצע דה-רנדומיזציה לבעיות קאנוניות, ועבור חישוב מוגבל-זכרון הבחירה של בעיית הקשירות במגוון תצורות היא מאוד מתבקשת. תוצאה קלאסית של סאביץ׳ [Sav70] מראה שניתן לפתור את בעיית הקשירות בגרפים מכוונים ב- L^2, מה שמראה ש- $NL \subseteq L^2$. התקדמות משמעותית נעשתה ע״י אלליונאס ואחרים [AKL+79] שהציעו אלגוריתם הסתברותי טבעי לפתרון בעית הקשירות בגרפים לא מכוונים.

תוך שימוש במכפלת הזיג-זג [RVW02], ריינגולד [Rei08] ביצע דה-רנדומיזציה לבעיית הקשירות בגרפים לא מכוונים ע״י הפיכה של גרף הקלט לגרף מרחיב בעל דרגה קבועה. הטכניקה הנהדרת של ריינגולד גוררת גם סדרת הילוכים אוניברסלית על גרפים לא מכוונים באורך פולינומי, אך רק תחת הגבלות מסוימות. יש לשים לב כי למרות שעבודתו של ריינגולד (ולאחר מכן של רוזנמן וודהאן [RV05]) משיגה דה-רנדומיזציה של העלאה בחזקה של גרפים לצורך פתרון בעית הקשירות, היא אינה, באופן ברור לפחות, גוררת אלגוריתם קירוב לחזקות של מטריצה בזכרון לוגריתמי. ריינגולד, טרוויסן וודהאן [RTV06] הרחיבו את תוצאתו של ריינגולד לגרפים מכוונים רגולריים (או, באופן כללי יותר, אוילריאניים) ובעיה שלמה חדשה ל- RL נמצאה – פתרון בעיית הקשירות בגרפים מכוונים עם זמן ערבוב פולינומי.

לעומת NL ו- RL, עבור המחלקה BPL לא היתה לנו בעיה טבעית שלמה, בפרט לא כזו שלא מנוסחת במונחי חזקות. כמו כן, לא היתה לנו מועמדת טבעית לבעיה ב- BPL שאינה כבר ב- L. ב- [DSTS17] נתנו בעיה טבעית ושלמה ל- BPL – קירוב הפער הספקטרלי של אופרטור סטוכסטי עם ערך-עצמי שני ממשי.[6] עשינו זאת ע״י המרת מכונת ה- BPL לגרף שכבות מכוון וניתוח הספקטרום שלו. קירוב הפער הספקטרלי הוא בעיה חשובה עם שימושים רבים, וחקר של בעיות באלגברה לינארית בהקשר של סיבוכיות זכרון עשוי להוליד תובנות חדשות בתחום, כפי שנדגים בהמשך.

אלגברה לינארית במעט זכרון ופתרון משוואות בלפלסיאן. מחלקת הסיבוכיות DET היא מחלקת כל השפות הניתנות לרדוקצית NC^1 לבעיה של חישוב הדטרמיננטה של מטריצה בעלת כניסות שלמות (ראו [Coo85] עבור ההגדרה המדויקת). מתקיים ש- $NL \subseteq DET \subseteq NC^2$. מתברר כי הרבה בעיות חשובות באלגברה לינארית, כגון היפוך מטריצות או לחילופין פתרון מערכת משוואות, הן ב- DET ולעתים שלמות עבור המחלקה. התוצאה $NL \subseteq DET$ היא של [Coo85] שהראה שבעיית הקשירות המכוונת ניתנת לרדוקציה לחישוב דטרמיננטה. העובדה ש- $DET \subseteq NC^2$ נובעת מהאלגוריתם של סאנקי [Csa76] לחישוב מקבילי של הדטרמיננטה. בנוסף, ידוע ש- $BPL \subseteq DET$.

חישוב חזקות גבוהות של מטריצה היא בעיה שלמה ל- DET. כמו כן, ניתן להראות כי **קירוב** חזקות של מטריצה סטוכסטית היא ב- BPL. ע״י המרת מכונת ה- BPL לאופרטור סטוכסטי A כך שההסתברות לעבור ממצב s למצב t תוך k צעדים היא $A^k[s,t]$ ניתן לראות שהבעיה אף שלמה ל- BPL.

להבנה טובה יותר של חישוב הסתברותי במעט זכרון, ניתן לשקול דווקא את חקר החוזק לכאורה של חישוב כזה, וכאמור, בעיות באלגברה לינארית הן מועמדות טבעיות. לפני מספר שנים, תא-שמע [TS13] הוכיח שניתן לקרב את הפירוק לערכים סינגולריים של מטריצה (ובכך גם את ההפכי שלה) עם שגיאה פולינומית-קטנה ב- BQL, האנלוג הקוונטי של BPL. ב- [DTS15b] ניגשנו לבעיה המתבקשת – דה-קוונטיזיציה של האלגוריתם

[6] התוצאות ב- [DSTS17] אינן מובאות בחיבור זה ומוזכרות כאן לטובת הבהרת הגישה הכללית.

חלק ב' – חישוב הסתברותי מוגבל-זכרון

אחת הבעיות החשובות בתורת הסיבוכיות היא, האם ניתן לחסוך במשאבי חישוב כגון זמן או זכרון ע"י שימוש באקראיות. נושא מרכזי בתחום מחקר זה הינו דה-רנדומיזציה של חישובים מוגבלי זכרון. חישוב הסתברותי מוגבל-זכרון הוא במרכזו של חלק זה של החיבור ולכן נגדיר כעת במדויק את מודל החישוב.

חישוב מוגבל-זכרון. למכונת טיורינג (להלן-'מ"ט') מוגבלת-זכרון דטרמיניסטית ישנם שלושה סרטים: סרט קלט (לקריאה בלבד), סרט עבודה (הניתן לקריאה ולכתיבה) וסרט פלט (לכתיבה חד-כיוונית בלבד). סיבוכיות הזכרון של המכונה היא מספר התאים בשימוש סרט העבודה. זמן הריצה של מ"ט מוגבלת-זכרון עם סיבוכיות זכרון $s(n) \geq \log n$ חסום ע"י $2^{O(s(n))}$. מ"ט מוגבלת-זכרון הסתברותית דומה לזו הדטרמיניסטית (ובפרט אנו דורשים שתעצור כעבור $2^{O(s(n))}$ צעדים) אך ביכולתה להטיל מטבעות אקראיים. הדרך הנהוגה לפרמל זאת היא ע"י הוספת סרט רביעי, סרט המטבעות האקראיים, שהוא לקריאה חד-כיוונית בלבד ומאותחל בהטלות מטבע אקראיות לחלוטין.

אנו נעסוק רק בחישוב עם שגיאה חסומה. נאמר ששפה מתקבלת ע"י מ"ט הסתברותית, אם לכל קלט בשפה הסתברות הקבלה של המכונה היא לפחות $2/3$ ולכל קלט שאינו בשפה הסתברות הקבלה היא לכל היותר $1/3$. כפי שקורה במקרים רבים, ניתן להגדיל את הפער בין שתי ההסתברויות כל עוד הפער ההתחלתי אינו זניח. נגדיר ששפה שייכת למחלקה $\mathsf{BPSPACE}(s(n))$ אם היא מתקבלת ע"י מ"ט מוגבלת-זכרון הסתברותית עם סיבוכיות זכרון $s(n)$. כמו כן, נגדיר $\mathsf{BPL} = \cup_c \mathsf{BPSPACE}(c \log n)$.

נגיד ששפה מתקבלת ע"י מ"ט הסתברותית עם שגיאה חד-צדדית, אם לכל קלט בשפה הסתברות הקבלה היא לפחות $1/2$ ולכל קלט שאינו בשפה, המכונה תמיד תדחה. נגדיר ששפה שייכת למחלקה $\mathsf{RSPACE}(s(n))$ אם היא מתקבלת ע"י מ"ט מוגבלת-זכרון הסתברותית עם שגיאה חד-צדדית וסיבוכיות זכרון $s(n)$. כמו כן, נסמן $\mathsf{RL} = \cup_c \mathsf{BPSPACE}(c \log n)$.

במחקר של חישוב הסתברותי מוגבל-זכרון, מספיק לבחון את המחלקות הנמוכות, ושם אנו יודעים ש-

$$\mathsf{NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{RL} \subseteq \mathsf{BPL} \subseteq \mathsf{NC}^2 \subseteq \mathsf{L}^2.$$

תחת הנחות קושי[4] אנו יודעים ש- $\mathsf{L} = \mathsf{RL} = \mathsf{BPL}$, ודה-רנדומיזציה מלאה, ללא הנחות, של חישוב הסתברותי בזכרון לוגריתמי מהווה אתגר חשוב בתורת הסיבוכיות מזה שנים רבות.

גישות מתחום הפסאודו-אקראיות לבעיה הניבו מגוון תוצאות ללא הנחות. מחולל הפסאודו-אקראיות[5] הראשון ל- BPL ניתן בעבודותיו החשובות של ניסן [Nis92, Nis94] שהוכיח כי $\mathsf{BPL} \subseteq \mathsf{DTISP}(\mathsf{poly}(n), O(\log^2 n))$, מה שגם נותן סדרת הילוכים אוניברסלית על גרפים לא מכוונים באורך קווזי-פולינומי. וריאנטים נוספים על המחולל של ניסן פותחו, ביניהם המחולל של אימפגליאצו, ניסן וויגדרסון [INW94] והמחולל של ניסן וצוקרמן [NZ96] שעושה שימוש במחלצי אקראיות ומראה שכל שפה המוכרעת ע"י מכונת BPL תוך שימוש במספר פולי-לוגריתמי של מטבעות כבר שייכת ל- L.

המרכיב המרכזי במחולל של ניסן הוא מחולל המרמה העלאה בריבוע של מטריצה סטוכסטית (לחילופין, מרמה שני צעדים של תכנית מסתעפת עם רוחב חסום). סאקס וז'ו [SZ99a] נתנו אלגוריתם משופר לקירוב חזקות של מטריצות סטוכסטיות ובעזרתו הוכיחו ש- $\mathsf{BPL} \subseteq \mathsf{L}^{3/2}$. במשפט אחד, נוכל להגיד כי סאקס וז'ו התגברו על התלויות בין רמות רקורסיה שונות במחולל של ניסן, מה שאפשר שימוש בפחות פונקציות גיבוב.

---

[4] קליוונס וון-מלקביק [KVM02] הוכיחו שאם קיימת שפה הניתנת להכרעה בזכרון לינארי ודורשת מעגלים בגודל אקספוננציאלי אז מתקיים ש- $\mathsf{L} = \mathsf{BPL}$.

[5] פורמלית, מחולל פסאודו-אקראיות עם שגיאה ε נגד משפחת פונקציות $\mathcal{F} \subseteq \{0,1\}^n \to \{0,1\}$ הוא פונקציה $G\colon \{0,1\}^d \to \{0,1\}^n$ כך שלכל $f \in \mathcal{F}$, $|\mathbb{E}[f(U_n)] - \mathbb{E}[G(U_d)]| \leq \varepsilon$.

מחלצים משני מקורות בעלי שגיאה נמוכה. למרות שהבניות האחרונות של מחלצים משני מקורות מגיעים קרוב מאוד לאנטרופיה לוגריתמית, החסרון המשותף לכולן הוא שהן אינן תומכות בשגיאה אקספוננציאלית-קטנה. במילים אחרות, זמן הריצה של המחלצים הללו אינו פולינומי ב- $\log(\frac{1}{\varepsilon})$ אלא במקרה הטוב, פולינומי ב- $\frac{1}{\varepsilon}$. בפרקים 4 ו- 5 אנו מציגים התקדמות מסוימת בטיפול בבעיית השגיאה.

בפרק 4 אנו ממשיכים לחקור את הקשר בין מחלצים משני מקורות למחלצים בלתי-חשילים ומציעים דרך לבנות מחלצים משני מקורות בעלי שגיאה נמוכה, התומכים בקצב אנטרופיה פולינומי, בהנתן מחלצים בלתי-חשילים טובים. הפרמטרים שאנו דורשים מהמחלצים הבלתי-חשילים כדי שהבנייה תעבוד תואמים (בנוחות) בניות לא מפורשות של מחלצים בלתי-חשילים, אך לצערנו כיום עדיין אין בניות מפורשות תואמות.

הבניה בפרק 4 משתמשת בשיטת דגימה שונה מאשר הבניות לאנטרופיות נמוכות ומייצרת מרחב מדגם קטן הרבה יותר עם הבטחה חד-צדדית בלבד. כתוצאה מכך אנו מצליחים להפטר מרכיב עיקרי שהיווה מחסום לקבלת שגיאה נמוכה. כלאחר יד, לא ברור האם בכלל קיימים דוגמים כאלו עם פרמטרים מתאימים. לשמחתנו, המפזרים של צוקרמן [Zuc07] (ראו גם בפרק 2.3.1) משיגים פרמטרים אופטימליים בטווח שמעניין אותנו.

בפרק 5 אנו משיגים בנייה מפורשת של **דוחס משני מקורות**, המחליש את הדרישה של מחלץ בכך שהתפלגות הפלט צריכה להיות קרובה להתפלגות עם הרבה אנטרופיה, אך לא בהכרח קרובה להתפלגות האוניפורמית. זמן הריצה של הדוחס שאנו בונים הוא $\text{poly}(n, \log \frac{1}{\varepsilon})$ והוא תומך באנטרופיות פולי-לוגריתמיות. הבניה שלנו מסתמכת על הכללת מושג הפונקציות החסינות, ששימוש בהן גורר שגיאה גבוהה, לפונקציות חסינות-אנטרופיה, שפולטות מספר סיביות גבוה ומאפשר שגיאה נמוכה. לדוחס שאנו בונים יש פער אנטרופיה קטן מאוד. ליתר דיוק, האנטרופיה בפלט בן m סיביות היא $m - o(\log \frac{1}{\varepsilon})$.

מחלצים משני מקורות לא מאוזנים ובניות נוספות. בפרק 6 אנו בונים מחלץ חדש משני מקורות לא מאוזנים ומשיגים פרמטרים כמעט אופטימליים. המחלץ שלנו פולט סיבית אחת עם שגיאה קבועה ממקור אחד באורך n ואנטרופיה $O(\log \log n)$ ומקור נוסף בלתי תלוי באורך $O(\log n)$ וקצב אנטרופיה קבוע קטן כרצוננו.

אנו מראים שמחלצים כאלו גוררים באופן מיידי מפזרים חזקים, כמעט אופטימליים, הפולטים סיבית אחת עם שגיאה נמוכה מאוד והפסד אנטרופיה נמוך גם כן (מפזרים יוגדרו במדויק בפרקים 2.3.1 ו- 6). בנוסף, אותם מחלצים נותנים בנייה של קודים מתקני שגיאות בינאריים הניתנים לפענוח ברשימה ממחיקות. הקודים שאנו בונים ניתנים לפענוח לאחר שחלק $1 - \varepsilon$ מהקואורדינטות נמחקו, ברשימה עם אורך כמעט אופטימלי של $\text{polylog}(1/\varepsilon)$ וקצב כמעט אופטימלי של $O(\varepsilon^{1+\delta})$ כאשר δ הוא קבוע קטן כרצוננו. זו הבניה הראשונה שמשיגה קצב טוב יותר מ- $O(\varepsilon^2)$, ובכך עונה על שאלה פתוחה שהוצגה כבר ב- [Gur04a, GI02, Gur04b]. הבניות בפרק 6 משלבות בניות חדשות בתחום הפסאודו-אקראיות וניתוח חדש ועדין לטיפול בשגיאות ובנושאי תלויות.

כפי שלרוב קורה בתחום הזה, קיימים קשרים רבים בין מגוון המחלצים לבין אובייקטים פסאודו-אקראיים אחרים, וכך גם קורה בחיבור זה. בפרק 2 נגדיר באופן פורמלי את מגוון המושגים והאובייקטים שנשתמש בהם, ואף נוכיח מספר טענות מקדימות.

אקראיות מפונקציות קשות, קודים לתיקון שגיאות ומחלצים בעלי גרעין. מעבר לשיפור בפרמטרים, טרוויסן ניתח את המחלץ שלו תוך שימוש בטכניקה חדשה – פרדיגמת השחזור, אשר הופכת את בעית חילוץ האקראיות לבעיה של שחזור מחרוזת ארוכה בהנתן מחרוזת עצה קצרה. מספר בניות אשר עשו שימוש בפרדיגמה זו פותחו בהמשך והציגו בניות מתוחכמות יותר עם פרמטרים טובים יותר, למשל [STSZ06, SU05, Uma03].

הבניות הללו השיגו פרמרטרים טובים, אך רק בעבודתיהם שלו לו, ריינגולד, ודהאן וויגדרסון [LRVW03] נבנו מחלצי אקראיות עם גרעין בעל אורך לוגריתמי. ב- 2007, גורוסוואמי, אומנס וודהאן [GUV09] הציגו בנייה ישירה ואלגנטית של מחלצים כמעט-אופטימליים עם תלות טובה יותר בשגיאה המבוססת על הקודים מתקני השגיאות של פרוורש וורדי. מספר בניות נוספות ניתנו בהמשך [DW11, DKSS13, TSU12] שהשיגו הפסד אנטרופיה קטן יותר.

מחלצים משני מקורות. השגת בניות מפורשות של מחלצים משני מקורות התגלתה כמאתגרת יותר.[3] המכפלה הפנימית עובדת היטב כאשר האנטרופיה k גדולה מ- $n/2$. בורגן [Bou05] נתן בנייה התומכת באנטרופיה $k = (1/2-\alpha)n$ עבור $\alpha > 0$ קבועה וקטנה. בשל הקושי לבנות מחלצים משני מקורות התומכים באנטרופיות נמוכות, מאמץ רב הושקע בבניית מחלצים ממספר מקורות בלתי-תלויים עבור אנטרופיות קטנות ומספר מקורות קטן ככל הניתן. ראו, למשל, אצל [BIW06, Rao09a, Li11, Li13a, Li13b, Li15b].

שני עשורים לאחר התוצאה של בורגן, הצליחו צ׳אטופדהייאי וצוקרמן [CZ16] להוריד בצורה דרסטית את האנטרופיה, ונתנו בנייה מפורשת של מחלץ משני מקורות לאנטרופיה פולי-לוגריתמית! בנייתם פורצת הדרך משתמשת ברעיונות מעבודות קודמות [Rao09a, Li15b] ועושה שימוש במחלצים בלתי-חשילים כמרכיב מרכזי. מחלצים בלתי-חשילים, שהוגדרו לראשונה ע״י דודיס ווויקס [DW09] בהקשרים של הגברת פרטיות, מקשיחים את הדרישה ממחלצים בעלי גרעין. מחלצים בלתי-חשילים הם מחלצים בעלי גרעין ובנוסף לדרישה שהתפלגות הפלט תהיה קרובה להתפלגות האוניפורמית, היא צריכה להיות קרובה להתפלגות האוניפורמית גם בהנתן הפלט של המחלץ על גרעינים שונים שנבחרו ע״י יריב כל-יכול (ראו את ההגדרה המדויקת בפרק 2.4). מספר שיפורים לתוצאה של צ׳אטופדהייאי וצוקרמן פורסמו זמן קצר לאחר מכן [Mek17, Li16]. כהן ושולמן [CS16] הבחינו כי כל הבניות הנ״ל תומכות באופן אינהרנטי רק באנטרופיות פולי-לוגריתמיות, והצליחו לתת את הבניה הראשונה של מחלץ ממספר מקורות התומך באנטרופיה כמעט לוגריתמית. צ׳אטופדהייאי ולי [CL16] הורידו את מספר המקורות בבנייה לקבוע, ולאחר מכן כהן [Coh16b] הצליח להוריד את מספר המקורות לחמישה.

מחלצים משני מקורות התומכים באנטרופיה כמעט לוגריתמית. בפרק 3 אנו לוקחים צעד משמעותי קדימה בדרך למחלץ אופטימלי משני מקורות ומציגים את הבניה המפורשת הראשונה התומכת באנטרופיה כמעט לוגריתמית. הבנייה ב- [CZ16] הציגה רדוקציה ממחלצים משני מקורות למלחצים בלתי-חשילים, אך גם אם נניח קיום מפורש של מחלצים בלתי-חשילים אופטימליים, הבנייה שלהם עדיין תתן אנטרופיה $\text{polylog}(n)$ ולא $O(\log n)$ כפי שהיינו רוצים.

התוצאה שלנו בחיבור זה מרחיבה את הסכימה של [CZ16] ותומכת באנטרופיות נמוכות יותר ע״י בניה של רדוקציה למחלצים בלתי-חשילים שהיא גם **משמרת אנטרופיה**. בקצרה, הבנייה ב- [CZ16] משתמשת במחלץ בעל גרעין כדוגם, ואנו נראה שניתן להחליש את דרישת הדגימה הקלאסית ולהשתמש בסוג מסוים של דוחס אקראיות המשיג דגימה עם שגיאה כפלית. אנו נבנה דוחס כזה, המשיג פער אנטרופיה נמוך ואורך גרעין טוב יותר משל דוגם אופטימלי. הבנייה של דוחס האקראיות משתמשת בטכניקת הורדת השגיאה של [RRV99]. מאז שבנייתנו פורסמה, מחלצים בלתי-חשילים טובים יותר נבנו [Coh16d, Li17, Li18] ותוך שימוש ברדוקציה המשופרת, קיומם גורר מחלצים משני מקורות התומכים באנטרופיה $O(\log n \frac{\log\log n}{\log\log\log n})$.

[3] עבודה רבה הושקעה בבנייה מפורשת של האובייקט החלש יותר, של מפזרים משני מקורות, הגוררים בנייה מפורשת של גרפי רמזי. נרחיב על כך בפרק 2.5.2.

תקציר

חלק א׳ – מחלצים משני מקורות ובניות נוספות

הבעיה של זיקוק אקראיות ממקורות חלשים עלתה עוד בעבודתו של וון-נוימן [Neu51]. באופן לא פורמלי, מה שהיינו רוצים הוא מחלץ אקראיות – אלגוריתם המייצר סיביות אקראיות לחלוטין ממקור בעל אקראיות פגומה. מחלצי אקראיות התבררו כחשובים הרבה מעבר למשימת הזיקוק הנ״ל, וכיום הם נמצאים במגוון תחומים כגון קודים לתיקון שגיאות, קריפטוגרפיה, קומבינטוריקה והוכחת חסמים תחתונים (ראו [Wig09 ,Sha02] והפניות נוספות שם).

באופן אידיאלי, מחלץ אקראיות היה מוגדר כפונקציה $\mathrm{Ext}\colon \{0,1\}^n \to \{0,1\}^m$ בעלת התכונה שלכל מקור X בעל מספיק אנטרופיה, ההתפלגות $\mathrm{Ext}(X)$ היא ε-קרובה להתפלגות האוניפורמלית מעל $\{0,1\}^m$ במרחק סטטיסטי, ואת זה נסמן כ- $\mathrm{Ext}(X) \approx_\varepsilon U_m$.

ואכן, ממקורות בעלי **מבניות** מסוימת ניתן לחלץ אקראיות בצורה דטרמיניסטית. למשל, פונקציה Ext כנ״ל קיימת עבור מקורות קובעי-סיביות (למשל, [CS15 ,Gab11 ,KZ06])، מקורות אפיניים (למקצת העבודות, ראו [Li16 ,Yeh11 ,Bou07])، ומקורות הניתנים לדגימה [Vio14 ,TV00]. יחד עם זאת, עבור מקורות כלליים בעלי אקראיות חלשה, כשההבטחה היחידה היא שאין בהתפלגותם איברים כבדים, לא ניתן לבצע חילוץ אקראיות דטרמיניסטי.

כדי להווכח בטענה זו, תחילה נגדיר את מושגי היסוד. אנו ממדלים **מקור חלש** כמשתנה אקראי X שלצורך נוחות נניח כי הוא מתפלג מעל $\{0,1\}^n$. המדד הסטנדרטי למידת האקראיות ב- X הוא האנטרופיה (ואנו נשתמש ב- min-entropy) שהיא ה- k המקסימלי עבורו לא ניתן לנחש את הערך של X בהסתברות טובה יותר מ- 2^{-k}. באופן שקול, האנטרופיה ב- X היא ה- k המקסימלי עבורו $\Pr[X = x] \leq 2^{-k}$ לכל x בתומך של X. עבור k כנ״ל, נגיד ש- X הוא (n,k) מקור, או פשוט k-מקור. כעת, נקבע פונקציה כלשהי $\mathrm{Ext}\colon \{0,1\}^n \to \{0,1\}$ ונניח בלי הגבלת הכלליות ש- $|\mathrm{Ext}^{-1}(0)| \geq |\mathrm{Ext}^{-1}(1)|$. יהא X המקור שמפולג אוניפורמית מעל $\mathrm{Ext}^{-1}(0)$. האנטרופיה ב- X היא גבוהה, לפחות $n-1$, אך $\mathrm{Ext}(X)$ הוא בבירור קבוע.

לאור העובדה שלא ניתן לחלץ אקראיות באופן דטרמיניסטי ממקורות חלשים כלליים, הוצעו מספר וריאנטים של מחלצי אקראיות המהווים רלקסציה של אותה הגדרה לא ישיגה. ב- **מחלצים בעלי גרעין** אנו מאפשרים למחלץ לקבל מחרוזת קצרה נוספת הנדגמת מהתפלגות אוניפורמית ובלתי תלויה במקור החלש. פורמלית, מחלץ בעל גרעין הוא פונקציה $\mathrm{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ המקבלת כקלט דגימה $x \in \{0,1\}^n$ מ- k-מקור X ומחרוזת $y \in \{0,1\}^d$ הנדגמת בצורה אוניפורמית ובלתי-תלויה, ופולטת $\mathrm{Ext}(x,y)$ כך שהתפלגות הפלט מקיימת $\mathrm{Ext}(X, U_d) \approx_\varepsilon U_m$.

וריאנט נוסף למלחץ ממקורות חלשים הוא **מחלץ משני מקורות** (או ממספר רב יותר של מקורות) בלתי תלויים, ונגיד שהפונקציה $\mathrm{2Ext}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ היא מחלץ משני מקורות לאנטרופיה k אם לכל שני מקורות בלתי תלויים X ו- Y מתקיים ש- $\mathrm{2Ext}(X,Y) \approx_\varepsilon U_m$. בבנייה לא מפורשת, k יכול להיות קטן עד כדי $\log n + 2\log(\frac{1}{\varepsilon}) + O(1)$.

מחלצים בעלי גרעין. בניית מחלצים בעלי גרעין היתה מוקד למחקר ענף במשך שלושה עשורים. במקור, מחלצים בעלי גרעין הוגדרו בהקשרי דה-רנדומיזציה של חישובים הסתברותיים מוגבלי-זכרון (ראו [NZ96] ומגוון עבודות המשך) וכדרך לסמלץ אלגוריתמים הסתברותיים ע״י מקורות חלשים [NZ96]. מאז, הם בשימוש רב בתיאוריה של מדעי המחשב ומהווים לעתים רכיב בבנייה של אובייקטים קומבינטוריים אחרים. נראה דוגמאות לכך בחיבור זה.

בניות מוקדמות של מחלצים בעלי גרעין התבססו על אי-תלות מוגבלת [SZ99b ,ILL89] והמרות מבניות שונות [SSZ98, TS96, NTS99]. טרוויסן [Tre01] ביצע פריצת דרך בתחום ע״י הבנת הקשר בין מחוללי פסאודו-

תמצית

תפקידה של אקראיות בחישובים היא בעיה יסודית וחשובה בתיאוריה של מדעי המחשב, הן במודל המוגבל חישובית (האם לכל אלגוריתם הסתברותי מוגבל-זמן או מוגבל-זכרון קיים אלגוריתם דטרמיניסטי שקול?) והן במודל הלא מוגבל חישובית (כיצד ניתן לזקק אקראיות מושלמת ממקורות בעלי אקראיות חלשה?). בעבודה זו נדון בבעיות בחילוץ אקראיות ובחישוב הסתברותי מוגבל-זכרון.

בחלק הראשון של החיבור נדון בחילוץ אקראיות ממספר מקורות חלשים בלתי תלויים, נושא הנחקר לעומק בעשורים האחרונים. לאחרונה פורסמו עבודות רבות בתחום ובפסגתן בנייה מפורשת של מחלץ אקראיות משני מקורות התומך באנטרופיה פולי-לוגריתמית [CZ16] ואיתה סט של כלים ואובייקטים חדשים בתחום. בנייה זו פורצת הדרך של צ׳אטופדהייאי וצוקרמן עשתה שימוש במחלצים בלתי-חשילים כרכיב מרכזי, אך בנייתם אינה אופטימלית גם לו היינו משתמשים במחלצים בלתי-חשילים אופטימליים. בחיבור זה אנו מציגים בנייה מפורשת של מחלץ אקראיות משני מקורות התומך באנטרופיה **כמעט לוגריתמית** ע״י פיתוח רדוקציה משמרת אנטרופיה למחלצים בלתי-חשילים המשתמשת בשיטת דגימה חדשה.

הבניות הנ״ל תומכות באנטרופיות נמוכות מאוד, אך אינן משיגות שגיאה אקספוננציאלית קטנה. בעבודה זו אנו משיגים התקדמות מסוימת לקראת מחלצים עם שגיאה נמוכה ונותנים שתי בניות חדשות. הבנייה הראשונה מניחה קיום של מחלצים בלתי-חשילים טובים, ובעזרתם משיגה מחלצים משני מקורות עם שגיאה נמוכה, התומכים בקצב אנטרופיה פולינומי. הבנייה השנייה היא בנייה מפורשת, אך משיגה רק דוחס אקראיות, ולא מחלץ. לדוחס האקראיות שגיאה נמוכה, הוא תומך באנטרופיה פולי-לוגריתמית ומשיג פער אנטרופיה קטן מאוד.

בסיום החלק הראשון של החיבור נציג בנייה חדשה, כמעט אופטימלית, של מחלץ אקראיות משני מקורות לא מאוזנים. בעזרת מחלצים אלו אנו משיגים גם מפזרי אקראיות חזקים הפולטים סיבית אחת עם שגיאה נמוכה, אורך גרעין קרוב לאופטימלי והפסד אנטרופיה קרוב לאופטימלי. בנייה זו משיגה פרמטרים טובים יותר מאלו של מחלצי אקראיות אופטימליים, וזו אחת מהדוגמאות הבודדות לכך. מנקודת המבט של קודים מתקני שגיאות, אנו מקבלים קודים בינאריים הניתנים לפענוח ברשימה ממחיקות, עם אורך רשימה קרוב לאופטימלי וקצב קרוב לאופטימלי.

בחלק השני של החיבור נדון בחישוב מוגבל-זכרון והקשרו לבעיות באלגברה לינארית, ונראה שבכדי לבצע דה-רנדומיזציה של סכימות קירוב מוגבלות-זכרון (למשל, עבור היפוך מטריצות) די לבצע דה-רנדומיזציה למחלקות ההכרעה המתאימות. אנו נעסוק גם בסיבוכיות זכרון של קירוב פתרון של מערכת משוואות לינאריות הניתנות ע״י לפלסיאן של גרף. לבעיה זו ניתן אלגוריתם הסתברותי הדורש זכרון לוגריתמי בלבד, גם עבור גרפים מכוונים בעלי זמן ערבוב פולינומי.

עבודות אלו, המתלוות לעבודות אחרות בתחום, חושפות את הקשר בין חישוב מוגבל-זכרון לבעיות באלגברה לינארית, שבו מחלקות הסיבוכיות השונות (לדוגמא, חישוב הסתברותי בזכרון לוגריתמי או חישוב קוונטי בזכרון לוגריתמי) מאופיינות ע״י בעיות אלגבריות, כאשר ההבדל נעוץ בסוג האופרטורים עבורם הבעיות מוגדרות.

חיבור זה מבוסס על המאמרים [DTS15a, DLGTS17, BACD$^+$18, BADTS18, BACDTS18] והיא פרי של שיתוף פעולה עם אברהם בן-ארויה, אישן צ׳אטופדהייאי, גיל כהן, פרנסואה לה-גל, שין לי ואמנון תא-שמע.

אוניברסיטת תל אביב

הפקולטה למדעים מדוייקים ע״ש ריימונד ובברלי סאקלר
ביה״ס למדעי המחשב ע״ש בלווטניק

מזקקי אקראיות וחישוב מוגבל-זכרון

חיבור לשם קבלת התואר
דוקטור לפילוסופיה
מאת

דין דורון

מנחה: פרופ׳ אמנון תא-שמע

הוגש לסנאט של אוניברסיטת תל אביב
אוגוסט 2018