On Extractors

Jon Rosario

March 12, 2024

‘

1 Introduction

Randomness is ubiquitous in computer science. It enables many applications, such as random number generation (RNG), simulations, cryptography, and randomized algorithms. In fields like machine learning/artificial intelligence, randomness enhances exploration in search algorithms and improves model training. Randomness also plays a significant role in ensuring the security of systems, as cryptographic keys and secure communication rely on unpredictable random values. In the study of algorithms, randomized forms of algorithms still provide much better guarantees on runtime than their deterministic counter-parts [Gas14].

Despite the numerous incredible ways that randomness has found usefulness in computer science, finding true sources of random bits is nearly impossible. In most cases, sources of randomness provided to computers are called **weak random sources**, containing biased or correlated bits. Randomness extractors were originally motivated by the need for generating nearly uniformly random bits from weak random sources. Creating nearly uniformly random bits from sources that are not very random seems like an impossible task, but in fact it is possible and has great theoretical implications.

Randomness extractors, or just extractors, are the name for the broad family of ideas on how to take a weakly random source and turn it into one that is nearly uniform distribution on m bits. Informally, extractors are deterministic functions that little bit of randomness into something very random. Note that this seems very similar to the definition of pseudorandom number generators (PRNGs), but they are not the same. Whereas the definition of extractors gives requires the output to be ϵ-close to uniform, the definition of PRNGs only requires outputs to be computationally indistinguishable from the uniform distribution, a significantly weaker notion. Secondly, extractors specify that a weakly random source should be used, but the same is not true for PRNGs [AB09].

The paper begins by defining crucial terms like entropy, min-entropy, and statistical distance, establishing a solid foundation in Section 2. It then delves into the different types of extractors: seedless, seeded, and strongly seeded, outlining their functionalities and properties in Section 3. Building on this framework, the Section 4 delves into the theoretical aspects, proving the existence of effective seeded extractors and offering an explicit construction utilizing 2-independent hash functions. Finally, we discuss further areas of exploration: connecting the world of extractors with error-correcting codes and introducing the concept of non-malleable extractors.

2 Definitions

Along with Shannon entropy, there are two other relevant measures of the amount of "randomness" or "surprise" in a random variable that are relevant to the discussion at hand. We define them below.

Definition 1. *(Entropy) Let X be a random variable and $\operatorname{supp}(X)$ be the support of a random variable. That is, the set of values of X that occur with non-zero probability. Then*

- *the **Shannon entropy** of X is*

$$H(X) = \sum_{x \in \operatorname{supp}(X)} p(x) \log 1/p(x) = E[-\log p(X)]. \tag{1}$$

- *the **min-entropy** of X is*
$$H_\infty(X) = \min_{x \in \operatorname{supp}(X)} \log 1/p(x). \tag{2}$$

- *the **Rényi entropy** of X (of order α) is*
$$H_\alpha(X) = \frac{1}{1-\alpha} \log(\sum_{x \in \operatorname{supp}(X)} p(x)^\alpha). \tag{3}$$

The Rényi entropy is a generalization of both the Shannon entropy and min-entropy. The limiting value of H_α as $\alpha \to 1$ is the Shannon entropy and the limiting value of H_α as $\alpha \to \infty$ is the min-entropy [EEB+04]. We take particular interest in $\alpha = 2$, which is also sometimes just referred to as the Rényi entropy or the collision entropy in other literature. We notate the collision entropy as H_2.

For each of the entropies $\mathcal{H} = \{H, H_\infty, H_2\}$, the following properties are satisfied.

Lemma 1. *For each of $H \in \mathcal{H}$ and random variables X and Y, the following properties are true [Vad12]:*

- *$H(X) \geq 0$ with equality if and only if X is supported by a single element.*
- *$H(X) \leq \log(|\operatorname{supp}(X)|)$ with equality if and only if X is uniform on $\operatorname{supp}(X)$.*
- *If X, Y are independent, then $H((X,Y)) = H(X) + H(Y)$.*
- *For all deterministic functions f, $H(X) \geq H(f(X))$.*
- *For every X, $H_\infty(X) \leq H_2(X) \leq H(X)$.*

These properties of entropy are somewhat expected and very natural. It is also not hard to see that we can think of the min-entropy as the largest non-negative real number $k \in \mathcal{R}_{\geq 0}$ such that $P[X = x] \leq 2^{-k}$ for every x in the support of X.

We also need to formalize the meaning of "close to the uniform distribution" in order to properly describe extractors. We use this by using a metric on on random variables.

Definition 2. *(Variation Distance) For random variables X, Y taking values in $\mathcal{U} = \operatorname{supp}(X) \bigcup \operatorname{supp}(Y)$, the **variation distance** is $\Delta(X,Y) = \max_{E \subset \mathcal{U}} |P_X[E] - P_Y[E]|$. We say X and Y are ϵ-close if $\Delta(X,Y) \leq \epsilon$.*

Informally, this is just the maximum difference in probabilities between the two distributions given that the same event happens. Similarly, there are certain properties that we expect from a good measure of variation distance between random variables.

Lemma 2. *For random variables X, Y, Z,*

- *$\Delta(X,Y) \geq 0$ with equality if and only if $X = Y$.*
- *$\Delta(X,Y) \leq 1$ with equality if and only if $\operatorname{supp}(X)$ and $\operatorname{supp}(Y)$ are disjoint.*
- *$\Delta(X,Y) = \Delta(Y,X)$.*
- *$\Delta(X,Z) \leq \Delta(X,Y) + \Delta(Y,Z)$, showing that Δ indeed is a metric on random variables.*
- *$\Delta(X,Y) = 1/2 \sum_{a \in \mathcal{U}} |P_X(a) - P_Y(a)|$*

In the literature, it is common to define the statistical variation as $1/2 \sum_{a \in \mathcal{U}} |P_X(a) - P_Y(a)|$. However, the last property of Lemma 2 shows that these are indeed equivalent.

Now we introduce random variables using the terminology used in the study of randomness extractors.

Definition 3. *If X is a distribution over $\{0,1\}^n$, with min-entropy $H_\infty(X) \geq k$, then it is called a $(\mathbf{n}, \mathbf{k})$-**source**.*

Example 1. Consider a series of coin tosses with a biased coin. Let X be the series of n independent coin tosses, where each flip results in a 1 with probability $\delta < 1/2$ and results in a 0 with probability $1 - \delta$. The min-entropy $H_\infty(X)$ can be shown to be $n \log \frac{1}{1-\delta}$ Therefore, X is a $(n, n \log \frac{1}{1-\delta})$ source [AB09].

Example 2. We can also consider Santha-Vazirani sources [SV84]. Let X be a series of n bits and $\delta < 1/2$. X also has the property that for all $i \in [n]$:

$$\delta \leq P[X_i = 1 | X_1 = x_1, ..., X_{i-1} = x_{i-1}] \leq 1 - \delta. \tag{4}$$

In other words, the probability the i-th bit of X being 1 even when given the knowledge of all previous bits is bounded between δ and $1-\delta$. Then the min-entropy $H_\infty(X) \geq H(\delta)n$. X is also an $(n, nH(\delta))$-source.

3 Seedless and Seeded Extractors

We now introduce the weakest class of extractors.

Definition 4. *Let $\mathcal{X}$ be a family of sources with* $\mathrm{supp}(X) = \{0,1\}^n$ *for $X \in \mathcal{X}$. We say a function $Ext : \{0,1\}^n \rightarrow \{0,1\}^m$ is a **seedless extractor** for $\mathcal{X}$ with error ϵ if for all $X \in \mathcal{X}$,*

$$\Delta(Ext(X), U_m) \leq \epsilon, \tag{5}$$

where U_m is the uniform distribution on m bits.

Example 3. In Example 1, how could one possibly transform this biased $(n, nH(\delta))$-source into a source of uniformly random bits? Von Neumann proposed a solution which is widely known to be one of the first and simplest (seedless) explicit extractors. The idea is to pair the results of consecutive coin flips. Then for each pair, output 0 if the outcome was 01, 1 if the outcome was 10, and skip the pair otherwise. This can be shown to produce a uniformly random distribution of bits, thus being a seedless extractor [VN63].

Unfortunately, seedless extractors are quite limited in power. In response to this limitation, Nisan and Zuckerman introduced the notion of extractors with seeds.

Definition 5. *We say a function $Ext : \{0,1\}^n \times \{0,1\}^t \rightarrow \{0,1\}^m$ is a **seeded** $(\mathbf{k}, \epsilon)$ **extractor** for any (n, k)-source X if:*

$$\Delta(Ext(X, U_t), U_m) \leq \epsilon, \tag{6}$$

*We call the second part of the input the seed of the extractor [NZ96]. We call a seeded (k, ϵ) extractor a **strongly seeded** $(\mathbf{k}, \epsilon)$ **extractor** if for any (n, k)-source X:*

$$\Delta((U_t, Ext(X, U_t)), (U_t, U_m)) \leq \epsilon. \tag{7}$$

Intuitively, strongly seeded extractors need to extract randomness independently of the seed.

A good point is that it seems like extractors are pointless if we require access to a truly random t bits in the first place. Indeed, in the case where $t \geq m$, trivial extractors exist–they just take its output from the uniformly random seed. However, we are interested in the case where $t << m$. Besides the original motivation of searching for ways to sufficiently randomize biased sources of randomness, extractors are also of practical interest to simulate what's known as the **bounded-error probabilistic polynomial time (BPP)** complexity class. This is a class of problems which can be solved efficiently with randomized algorithms with a probability of correctness of at least $1/2 + c$ for some $c > 0$. For the purposes of simulating algorithms for problems in **BPP**, we would like to be able to generate m uniformly random bits for use in the algorithm using a weakly random source X. Suppose there exists an such a seeded extractor with seed size $t = O(\log n)$. Then we could just efficiently simulate the algorithm by enumerating over the extractor with all $2^t = O(n)$ possible inputs for the seed.

We will now prove the existence of good seeded extractors. By "good", we mean that the seed size is small ($t << m$). However, to do this, we use a similar theorem that states the existence of seedless extractors a certain class of special sources.

Definition 6. *We say X is a **flat** (n,k)**-source** if for any $x \in \{0,1\}^n$, $P[X=x]=0$ or $P[X=x]=2^{-k}$.*

The notion of a source is very broad and in general can be hard to use in proofs. Fortunately, flat sources are the saving grace! These sources correspond exactly to a uniform distribution on a subset $S \subseteq \{0,1\}^n$ with $|S|=2^k$. Furthermore, they are very powerful by the following lemma.

Lemma 3. *For any source X, X is an (n,k)-source if and only if it is a distribution $\mathcal{D}$ on flat (n,k)-sources $X_1, X_2, \ldots$. Plainly, a sample of X corresponds to picking one of the X_i according to $\mathcal{D}$ and then drawing a sample from X_i [Vad12].*

This enables us to show the existence of seedless extractors for the class of flat sources of min-entropy $H_\infty \geq k$.

Theorem 1. *For every flat (n,k)-source X and $m \leq n$, there exists a seedless extractor $Ext : \{0,1\}^n \to \{0,1\}^m$ with*

- *$\Delta(Ext(X), U_m) \leq \epsilon$ and*
- *$m = k - 2\log 1/\epsilon - O(1)$.*

This extractor can be chosen randomly. In this case, the probability that extractor is ϵ-close to U_m is $1 - 2^{-\Omega(2^k \epsilon^2)}$ [Sha11, Vad12].

The proof of this theorem can be shown probabilistically using Lemma 3 and the Chernoff bounds. Perhaps more interestingly is that Theorem 1 almost directly implies the existence of good seeded extractors.

Theorem 2. *For every (n,k)-source X and $\epsilon > 0$, there exists a seeded (k,ϵ) extractor $Ext : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ with*

- *$\Delta(Ext(X), U_m) \leq \epsilon$,*
- *$m = k + t - 2\log(1/\epsilon) - O(1)$, and*
- *$t = \log(n-k) + 2\log(1/\epsilon) + O(1)$.*

This theorem is known to be optimal in respect to the relationship between error probability ϵ, output length m, and seed length t. The idea of the proof is to first apply Lemma 3 to simplify the analysis. Then, consider a new source $X' = (X, U_t)$ with min-entropy $k+t$. By applying Theorem 1, we know the probability of this (partially) randomly constructed extractor failing. All that is left to show is that the probability of this extractor failing for all distribution of flat sources is less than 1.

Proof. We choose an candidate extractor function at random with $m = k + t - 2\log(1/\epsilon) - O(1)$. By Lemma 3, we only need to show that this candidate extractor function works for all flat (n,k)-sources. Using this lemma, decomposing an arbitrary (n,k)-source into multiple flat (n,k)-sources means the probability of the random extractor failing is at most the probability of random extractor failing for a fixed flat (n,k)-source times the number of flat (n,k)-sources.

Next, we note that if X is a (n,k)-source, then (X, U_t) is a flat $(n+t, k+t)$-source. By Theorem 1, the probability of such a randomly constructed extractor failing on a fixed flat (n,k)-source is at most $2^{-\Omega(2^{k+t}\epsilon^2)} = 2^{-\Omega(2^k 2^t \epsilon^2)}$. Thus, the probability of failing in general is just

$$\binom{2^n}{2^k} 2^{-\Omega(2^k 2^t \epsilon^2)} \leq \left(\frac{2^n e}{2^k}\right)^{2^k} 2^{-\Omega(2^k 2^t \epsilon^2)} < 1 \tag{8}$$

where the first inequality follows from a common bound on the binomial coefficients, and the second inequality is true when $2^t \epsilon^2 \geq O(\log 2^{n-k} e)$. Rearranging this equation shows the necessary bound on t. Note that this proof can also be extended to work for strongly seeded extractors. □

This probabilistic argument shows the existence of good extractors, but does not explicitly show how to construct such an extractor.

4 Explicit Extractors

Since the introduction of extractors in [NZ96], explicit constructors of extractors whose bounds match those given in Theorem 2 have been hard to find. Fortunately, now there are explicit constructions of seeded extractors that asymptotically match the probabilistic construction of Theorem 2, the first of which was given by Lu, Reingold, Vadhan, and Wigderson [LRVW03].

These extractors are known to be nearly optimal–the only problem being that the error ϵ needs to be fairly large. The work of Guruswami, Umans, and Vadhan improves upon this construction in [GUV07] by giving a new, simpler construction based on the theory expander graphs. Furthermore, this work achieves a better error parameter ϵ.

However, the construction is still far too complicated to fit in this paper comfortably. As a result, we introduce a less-than-optimal construction employing 2-independent hash functions to enhance comprehension.

Definition 7. *A collection $\mathcal{H}$ of functions $h : \{0,1\}^n \to \{0,1\}^m$ ($H \in \mathcal{H}$) is* 2***-independent****, or* ***pairwise independent****, if for every $x \neq x' \in \{0,1\}^n$ and $y, y' \in \{0,1\}^m$,*

$$P_{h\in\mathcal{H}}[h(x) = y \wedge h(x') = y'] = \frac{1}{2^{2m}}.$$

That is to say, if we can uniformly at random select a hash function $h \in \mathcal{H}$, then the random variables $h(x)$ and $h(y)$ are uniformly distributed and pairwise independent. There are known constructions of collections $\mathcal{H}$ which have this property. The Carter and Wegman construction in [CW79] is popular, since we are able to completely specify a hash function $h \in \mathcal{H}$ with a string of length $n+m$. We will use this idea to form extractors.

Definition 8. *Suppose we have an (n,k)-source X and we draw from this source twice to get two length n binary strings x, y. The* ***collision probability*** *of X is $P[x = y] = \sum_{x,y} p(x,y) = \sum_{x\in\text{supp}(X)} p_x^2$.*

Now, we state a lemma upper bounding the collision probability for (n,k)-sources, based on the interpretation of min-entropy H_∞ that each draw from X should produce at least k bits of randomness.

Lemma 4. *If X is an (n,k)-source, then the collision probability is at most 2^k.*

Proof. Left as an exercise for the reader. Hint: apply Definition 1 and Lemma 1. □

Using this lemma, we can prove a very important lemma in cryptography first stated by Russell Impagliazzo, Leonid Levin, and Michael Luby in [ILL89] that tells us something about how many uniformly random bits we can extract from a source X.

Lemma 5. *(Leftover Hash Lemma) Let $m = k - 2\log(1/\epsilon)$. For every (n,k)-source X:*

$$\Delta((H(X), H), (U_n, H)) \leq \epsilon \tag{9}$$

where H is a hash function selected uniformly at random from a pairwise independent collection $\mathcal{H}$ of hash functions $h : \{0,1\}^n \to \{0,1\}^m$.

Proof. Conceptually, the proof can be broken into three parts. The first part is a direct application of Lemma 4. The second part is to show that this implies that the output of the extractor is close to uniform under the L^2 norm. The variation distance as defined above uses the L^1 norm, so the last part is to show that the output of the extractor is also closed to uniform under the L^1 norm by a general inequality relating the two. We refer the reader to [Vad12] for a full proof. □

For explicitness, the extractor $Ext : \{0,1\}^n \times \{0,1\}^{n+m} \to \{0,1\}^m$ is the function which takes in as input n bits from a source X and a truly random hash function h described by a string of length $n+m$, and outputs m bits close to uniformly random, or $Ext(x,h) = h(x)$.

This extractor is not practical–the seed length is longer than the output length! However, it serves as a good starting point for building good extractors. Namely, the output length matches the theoretical bound in Theorem 2.

5 Discussion

Extractors, with their inherent theoretical appeal, have sparked a vibrant and diverse research landscape, with each direction yielding promising results. This section explores some of these promising directions. Firstly, extractors demonstrate a strong connection with other fields, such as error-correcting codes. We start by introducing a generalization of decodable codes.

Definition 9. *For $x, y \in \{0,1\}^n$, let $d(x,y)$ be Hamming distance between x and y. A function $C : \{0,1\}^n \to \{0,1\}^{2^t}$ is $(\mathbf{l}, \epsilon)$-**list decodable** if for every $z \in \{0,1\}^{2^t}$*

$$|\{x : d(C(x), z) \le (1/2 - \epsilon)2^d\}| \le l. \tag{10}$$

In other words, C represents a particular code for the strings in $\{0,1\}^n$. Then, if some noisy channel were to flip $1/2 - \epsilon$ of the indices of $C(x) \to z$, the receiver would still be able to figure out a list of at most l messages such that one of them is the original x. As disparate as it may seem, list decodable codes are exactly equivalent to strongly seeded extractors.

Theorem 3. *List decodable codes and strongly seeded extractors are equivalent [Tre01].*

- *If $Ext : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}$ is a strongly seeded (k, ϵ) extractor then $C(x)_y = Ext(x,y)$ is a $(2^k - 1, 2\epsilon)$-list decodable code.*
- *If $C : \{0,1\}^n \to \{0,1\}^{2^t}$ is a (l, ϵ)-list decodable code, then $Ext(x,y) = C(x)_y$ is a strongly seeded $(k, 2\epsilon)$-extractor for $k = \log l + \log(1/\epsilon) + 1$.*

Thanks to the original breakthrough presented in [Tre01], an entire unified theory of extractors and error-correcting codes exist.

Another exciting direction of seeded extractors occurs due to their usefulness in cryptographically secure settings. Consider the following scenario: we would like to communicate with a friend by using some shared randomly generated key. A natural protocol for this might be to use a weak source X, and a random seed y, then apply an extractor to produce a random key. Assuming both parties have access to the same weak source X, we might practically do this by sending our friend the seed y. After we apply an extractor, we both have access to the same random key R. However, this is obviously not secure–an adversary could intercept the message of the seed y, transform it to some other seed $f(y)$, and then send that to our friend.

Without delving much more into the cryptography of it, one could ask for a new type of extractor to somewhat combat this issue. Consider an extractor which takes a weak source X and a seed y and extracts some random bits R. However, even when an adversary gains the seed y and tampers with it to produce $f(y)$, the two values produced using each seed y and $f(y)$ results in two different values R and R' which are completely unrelated. Formally:

Definition 10. *A function $nmExt : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ is a **non-malleable extractor** if for any (n,k) source X and tampering function $f : \{0,1\}^t \to \{0,1\}^t$ with no fixed-points:*

$$\Delta((nmExt(X, U_t), nmExt(X, f(U_t)), U_t), (U_m, nmExt(X, f(U_t)), U_t)) \le \epsilon$$

This idea was only introduced fairly recently and were originally probabilistically shown to exist in [DW09], but already have already been proven to exist in [Li16, CL16, Li20]. This particular line of study of randomness extractors has been hugely successful, resulting in many other breakthroughs in cryptography.

There are many other forms of extractors only introduced in the past ten years that have already found much theoretical interest and practical interest that haven't been covered here. To list a few: multi-source extractors, seedless extractors, formulations of extractors as graph theoretic objects, and quantum-classical extractors. The reader is encouraged to continue reading on extractors, perhaps starting with this recent journal paper [Cha22].

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach.* Cambridge University Press, 2009.

[Cha22] Eshan Chattopadhyay. Recent advances in randomness extraction. *Entropy*, 24(7), 2022.

[CL16] Eshan Chattopadhyay and Xin Li. Explicit non-malleable extractors, multi-source extractors and almost optimal privacy amplification protocols, 2016.

[CW79] J.Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

[DW09] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. New York, NY, USA, 2009. Association for Computing Machinery.

[EEB$^+$04] Shannon Entropy, Renyi Entropy, Information P A Bromiley, N A Thacker, and Evelina Vassileva Bouhova-Thacker. Shannon entropy, renyi entropy, and information. 2004.

[Gas14] William Gasarch. Chapter five - classifying problems into complexity classes. volume 95 of *Advances in Computers*, pages 239–292. Elsevier, 2014.

[GUV07] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from parvaresh-vardy codes. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pages 96–108, 2007.

[ILL89] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, page 12–24, New York, NY, USA, 1989. Association for Computing Machinery.

[Li16] Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors, 2016.

[Li20] Xin Li. Non-malleable extractors and non-malleable codes: Partially optimal constructions. In *Proceedings of the 34th Computational Complexity Conference*, CCC '19, Dagstuhl, DEU, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[LRVW03] Chi-Jen Lu, Omer Reingold, Salil Vadhan, and Avi Wigderson. Extractors: Optimal up to constant factors. pages 602–611, 2003.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

[Sha11] Ronen Shaltiel. An introduction to randomness extractors. In Luca Aceto, Monika Henzinger, and Jiří Sgall, editors, *Automata, Languages and Programming*, pages 21–41, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[SV84] M. Santha and U.V. Vazirani. Generating quasi-random sequences from slightly-random sources. In *25th Annual Symposium onFoundations of Computer Science, 1984.*, pages 434–440, 1984.

[Tre01] Extractors and pseudorandom generators. *J. ACM*, 48(4):860–879, jul 2001.

[Vad12] S. Vadhan. *Pseudorandomness.* Foundations and Trends(r) in Theoretical Computer Science. Now Publishers, 2012.

[VN63] John Von Neumann. Various techniques used in connection with random digits. *John von Neumann, Collected Works*, 5:768–770, 1963.