Extractors for Turing-Machine Sources

Emanuele Viola*

Northeastern University, Boston MA 02115, USA
viola@ccs.neu.edu
http://www.ccs.neu.edu/home/viola/

Abstract. We obtain the first deterministic randomness extractors for n-bit sources with min-entropy $\geq n^{1-\alpha}$ generated (or sampled) by single-tape Turing machines running in time $n^{2-16\alpha}$, for all sufficiently small $\alpha > 0$. We also show that such machines cannot sample a uniform n-bit input to the Inner Product function together with the output.

The proofs combine a variant of the crossing-sequence technique by Hennie [SWCT 1965] with extractors for block sources, especially those by Chor and Goldreich [SICOMP 1988] and by Kamp, Rao, Vadhan, and Zuckerman [JCSS 2011].

Keywords: turing machine, independent source, deterministic randomness extractor, sampling lower bound, complexity of distributions.

1 Introduction

Turing machines may be the most studied model of computation even after decades of work on circuits. Following a first wave of worst-case lower bounds starting in the 60's (cf. [13]) and continuing to this date, researchers in the 90's have produced a second type of results. Specifically, Impagliazzo, Nisan, and Wigderson obtain in [14] average-case lower bounds and pseudorandom generators.

In this work we are interested in what we see as a third type of lower bounds: *sampling* lower bounds. We seek to understand what distributions can be sampled by randomized Turing machines (which take no input).

The first work on sampling complexity may be the one by Jerrum, Valiant, and Vazirani [15] who define sampling complexity classes and prove reductions among various problems. An unconditional communication complexity lower bound for sampling disjointness appears in the work [2] by Ambainis, Schulman, Ta-Shma, Vazirani, and Wigderson. Goldreich, Goldwasser, and Nussboim study the complexity of sampling in [11] as part of a general study of the implementation of huge random objects. Aaronson proves in [1] a connection between sampling and searching problems.

The complexity of sampling is being revisited in a series of recent works [26,19,9,25,6]. These works establish the first unconditional lower bounds for several computational models, such as bounded-depth circuits, and draw several new

* Supported by NSF grant CCF-0845003.

A. Gupta et al. (Eds.): APPROX/RANDOM 2012, LNCS 7408, pp. 663–671, 2012.

connections to problems in data structures, combinatorics, and randomness extractors. The connection to randomness extractors in particular makes progress along the research direction initiated by Trevisan and Vadhan in [24], and continued by Kamp, Rao, Vadhan, and Zuckerman in [16], which aims to construct deterministic randomness extractors for efficiently-samplable distributions.

1.1 Our Results

Our main result is an extractor for sources samplable by Turing machines running in subquadratic time. For clarity we first review randomized Turing machines.

In this work, Turing machines have exactly one read-write tape, infinite to the right only, with exactly one head on it. One may choose $\{0,1\}$ as tape alphabet. The tape is initially blank, that is, all zeros. In one time step, the machine reads the content of the cell, tosses a coin, and then writes the cell, updates the state, and moves the head to an adjacent location. Machines never halt, and we are only interested in a portion of their computation table. A $t \times t$ computation table is a $t \times t$ matrix corresponding to a valid computation according to such rules, with rows being configurations. Each entry specifies the content of the corresponding tape cell, whether the head is on that cell, and if so what is the current state and the current coin toss. Since we store the coin tosses in the entries, all $t \times t$ computation tables have equal probability 2^{-t}.

A Turing machine source on n bits running in time t is sampled as follows. First sample uniformly the $t \times t$ computation table. Then output the bottom left n tape bits.

Theorem 1 (Extractors for Turing-machine sources). *For all sufficiently small $\alpha > 0$,*
there is an explicit extractor $E : \{0,1\}^n \to \{0,1\}^m$ with output length $m = n^{\Omega(1)}$ and error $2^{-n^{\Omega(1)}}$ for n-bit sources with min-entropy $\geq k := n^{1-\alpha/16}$ that are sampled by Turing machines with $\leq 2^q := 2^{n^{\alpha/16}}$ states and running in time $\leq t := n^{2-\alpha}$.

The above theorem implies sampling lower bounds for somewhat complicated functions. The next one obtains one for the inner-product function IP.

Theorem 2 (Sampling lower bound for Turing machines). *For every $\alpha \in (0,1]$ and all sufficiently large even n*

no Turing machine with $\leq 2^q := 2^{n^{\alpha/2}}$ states and running in time $\leq t := n^{2-\alpha}$ can sample the distribution

$$(X_1, X_2, IP(X_1, X_2))$$

where X_1 and X_2 are uniform and independent over $\{0,1\}^{n/2}$.

Note that this result depends on the ordering of the input bits – if the bits of X_1 and X_2 are interleaved then a Turing machine can sample the distribution in linear time.

1.2 Overview of the Proofs

To prove our results we show that any Turing-machine source contains an independent source. More specifically, divide the n bits of the source into r blocks (or runs) of length ℓ separated by blocks of length b, as in Figure 1. We show that any Turing-machine source running in subquadratic time is a convex combination of sources $Y_1 Y_2 \cdots Y_r$ where the Y_i are independent, and each Y_i covers exactly one of the ℓ-bit blocks:

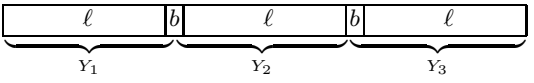

Fig. 1. Decomposition of Turing-machine source in $r = 3$ blocks (or runs) or ℓ bits separated by blocks of b bits

Lemma 1 (Turing-machine sources contain independent sources). *Let X be a Turing machine source on n bits running in time $t \geq n$ with 2^q states and min-entropy k.*

For any ℓ, b such that $(r-1)(\ell+b)+\ell = n$, X is a convex combination of $J \leq 2^{r \cdot O(q(\lg t)t/b)}$ n-bit sources S_j where each S_j is

$$S_j = Y_1 Y_2 \ldots Y_r,$$

where the Y_i are independent, and for every $i < r$ we have $\ell i + b(i-1) \leq |Y_1 Y_2 \ldots Y_i| \leq \ell i + bi$.

One can then extract using extractors for independent sources, developed in an exciting, ongoing line of research; see e.g. [22,8,3,4,21,16,20,7,5,18]. One gets different results depending on which extractors one uses. However, many of the available extractors for independent sources require a guarantee on the min-entropy of each source. By contrast, our given guarantee on the min-entropy of the Turing-machine source only translates into a guarantee on the *total* min-entropy of the independent sources. Thus for our extractor in Theorem 1 we use the extractors by Kamp, Rao, Vadhan, and Zuckerman [16] which only require that.

The sampling lower bound for IP in Theorem 2 is obtained by using instead the result by Chor and Goldreich that the inner product function $IP : \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}$ is a two-source extractor with error ϵ if the sum of the entropies of the two sources is $> \ell + 2\lg(1/\epsilon)$.[8]

We now elaborate on how we prove that any Turing-machine source contains an independent source. First, we introduce a variant of the classical crossing-sequence technique due to Hennie [12] that is suitable for sampling tasks. This allows us to sample the Turing-machine source by a one-way low-communication protocol among r players. This is explained in more detail below. Compared to previous simulations [17, §12] ours has the advantage of incurring no error. Another difference is that in our setting it is advantageous to have a large number

of players. (This is because the number of players corresponds to the number of independent blocks, and in general the more the independent blocks the easier the extraction.)

We then use the fact that a source sampled by a one-way low-communication protocol is a convex combination of independent sources. For 2 players, this fact originates from the work [2, §7] of Ambainis, Schulman, Ta-Shma, Vazirani, and Wigderson. Alternatively, one may view the sources sampled by such protocols as the extension of the source model in [16] where we output blocks instead of bits.

This concludes the high-level view of the proof. In the next paragraph we elaborate on how to sample a Turing-machine source by a low-communication protocol.

From Turing's Machines to Yao's Protocols. Let $T := (C_1, C_2, \ldots, C_t)$ be a distribution on $t \times t$ computation tables, where C_i represents the ith column of the table. We first describe an alternative way to sample T; then we explain how this alternative way can be implemented as a low-communication protocol.

The alternative way to sample T comes from the observation that the random variables $C_1, C_2, \ldots$ are a markov process (or chain). That is, conditioned on C_i, the random variable $C_{<i}$ of the columns before the ith is independent from the random variable $C_{>i}$ of the columns after the ith. The alternative way proceeds by sampling T from left to right one column at the time, each time conditioning only on the previous column (as opposed to the entire prefix). For example, one first samples $C_1 = c_1$, then samples $C_2 = c_2 | C_1 = c_1$, then samples $C_3 = c_3 | C_2 = c_2$, and so on. Let us call the resulting distribution $T^?$. To see that T and $T^?$ are the same distribution, note that after conditioning on a column $C_i = c_i$, T becomes a product distribution: the columns before i are independent from those after i. This holds because $T | C_i = c_i$ is uniform on its support (since each computation table has probability 2^{-t}), and by locality of computation: if $c_{<i}\ c_i\ c_{>i}$ and $c'_{<i}\ c_i\ c'_{>i}$ are in the support of $T | C_i = c_i$, then so is $c_{<i}\ c_i\ c'_{>i}$. It is now an exercise to show that for any transcript $t = (c_1, c_2, \ldots, c_t)$ we have $\Pr[T = t] = \Pr[T^? = t]$. The solution to the exercise follows.

$$\begin{aligned}
\Pr[T = t] &= \prod_i \Pr[C_i = c_i | C_{<i} = c_{<i}]; \\
\Pr[T^? = t] &= \prod_i \Pr[C_i = c_i | C_{i-1} = c_{i-1}] \\
&= \prod_i \frac{\Pr[C_i = c_i \wedge C_{<i-1} = c_{<i-1} | C_{i-1} = c_{i-1}]}{\Pr[C_{<i-1} = c_{<i-1} | C_{i-1} = c_{i-1}]} \\
&\qquad \text{(Since } T | C_{i-1} = c_{i-1} \text{ is product)} \\
&= \Pr[T = t].
\end{aligned}$$

We then exploit the above alternative way to sample T efficiently by a low-communication protocol among r players. Refer to Figure 1 for the parameters. The first player samples one column at a time. After an appropriate number ℓ

of columns, it looks for the first column that has a short description. By locality of computation, among b columns there must be one that corresponds to a tape cell that the Turing-machine head scans $\leq t/b$ times. Since modifications of a column only occur when the head scans it, this column can be described with about t/b bits, which is $< n$ for $t = n^{2-\alpha}$ and $b = n^{1-\alpha/2}$. The player can send this description to the next player, who can then continue the process.

2 Proofs

Proof (of Lemma 1). We prove this in two stages. In the first, more substantial stage we show how to sample the entire source X using a one-way low-communication protocol in which Player i outputs a sample covering Y_i but touching no Y_j for $j \neq i$. In the second stage we condition on the protocol's transcript.

We now proceed to the first stage. Let $T = (C_1, C_2, \ldots, C_t)$ be the uniform distribution over $t \times t$ computation tables.

P_1 starts sampling T from left to right, one column at the time. It stops at the first tape-cell index s_1 such that $\ell < s_1 \leq \ell + b$ and such that the sample c_{s_1} of C_{s_1} contains $\leq t/b$ states. Since each row only has the state in one cell, such an s_1 is guaranteed to exist. Because changes to tape contents only happen when the head is on that cell, this column can be described with

$$O(q(\lg t)t/b)$$

bits. The $\lg t$ term arises from specifying the times where the head is on that cell.

P_1 outputs the first s_1 output bits of the computation table. It then sends both the description of c_{s_1} and s_1 to P_2. This takes $O(q(\lg t)t/b) + O(\lg t) = O(q(\lg t)t/b)$ bits.

P_2 will then continue sampling the computation table from left to right one column at the time. It stops at the smallest tape-cell index s_2 such that $(\ell + b) + \ell < s_2 \leq 2(\ell + b)$ and such that the sample c_{s_2} of C_{s_2} contains $\leq t/b$ states. And so on.

This is the end of stage 1.

By conditioning on the communication, we can write the output distribution as a convex combination of $J \leq 2^{r \cdot O(q(\lg t)t/b)}$ distributions S_j. After conditioning on the communication, the players' output are independent and have a fixed length. Hence each S_j is a product distribution $S_j = Y_1 Y_2 \ldots Y_r$ where Y_i is the output of P_i. The bounds on the lengths of $Y_1 Y_2 \ldots Y_i$ follow by inspection.

The following standard claim bounds the entropy loss when selecting a distribution from a convex combination.

Claim (Entropy Loss in Convex Combo). Let D be a distribution with min-entropy k that is a convex combination of $J = 2^j$ distributions $D_1, D_2, \ldots, D_J$. Consider sampling D by first appropriately selecting an index $h \leq J$, and then sampling D_h. For every ϵ, the probability over the choice of h that D_h has min-entropy $\leq k - j - \lg(1/\epsilon)$ is $\leq \epsilon$.

Proof. Suppose the probability is $> \epsilon$. There is a $h \leq J$ that is picked with probability $> \epsilon/J$ such that D_h has min-entropy $\leq k - j - \lg(1/\epsilon)$. This means that there is some a such that $\Pr[D_h = a] \geq 1/2^{k-j-\lg(1/\epsilon)}$. But then $\Pr[D = a] > \epsilon/J \cdot 1/2^{k-j-\lg(1/\epsilon)} > 1/2^k$.

We use the following extractor.

Theorem 3 (Theorem 5.1 in [16]). *There is a constant $\beta > 0$ such that for every ℓ and $\delta \geq 1/\ell^\beta$ there is an explicit extractor for min-entropy $\geq \delta r \ell$ sources over $\left(\{0,1\}^\ell\right)^r$ such that the r blocks of ℓ bits are independent and with $r \geq 1/(\beta\delta^2)$, with output length $m = \ell^{\Omega(1)}$, and error $\epsilon = 2^{-\ell^{\Omega(1)}}$.*

Using the techniques in [10,23] one can derive a similar extractor where almost all the entropy is output, cf. [16, §7]. However we do not pursue this here.

We now prove our main extractor result.

Proof (of Theorem 1). For an α to be determined later, set $b := n^{1-\alpha/2}$ and $\ell := n^{1-\alpha/4}$. We assume w.l.o.g. that $\ell+b$ divides $n+b$. Note $r := (n+b)/(\ell+b) = \Theta(n^{\alpha/4})$.

Divide the n bits of the source into r runs of ℓ bits separated by $r-1$ runs of b bits. We apply the extractor from Theorem 3 to the r runs of ℓ bits.

By Lemma 1 we view the source as a convex combination of $J \leq 2^{O(rq(\lg t)t/b)}$ product sources S_j. By Claim 2 with $\epsilon := 2^{-k/2}$, if we choose a distribution in the combination, except with probability ϵ we obtain a distribution with min-entropy at least

$$\begin{aligned} &k - O(rq(\lg t)t/b) - \lg(1/\epsilon) \geq k/2 - O(rq(\lg t)t/b) \\ =&n^{1-\alpha/16}/2 - O(n^{\alpha/4+\alpha/16+1-\alpha/2} \lg n) = n^{1-\alpha/16}/2 - O(n^{1-3\alpha/16} \lg n) \\ \geq&\Omega(k). \end{aligned}$$

We assume this is the case and proceed.

By ignoring the $r-1$ runs of b bits, we drop $(r-1)b \leq O(n^{\alpha/4}n^{1-\alpha/2}) = O(n^{1-\alpha/4})$ bits. Since $k \geq n^{1-\alpha/16}$, the extractor is applied to a distribution of entropy that is still $\Omega(k)$.

Also, since we ignore the $r-1$ runs of b bits, the r runs of ℓ bits to which the extractor is applied are independent.

The parameter δ in theorem 3 is

$$\delta = \Theta(k/r\ell) = \Theta(k/n) = \Theta(1/n^{\alpha/16}).$$

We must have

$$\delta \geq 1/\ell^\beta = 1/n^{(1-\alpha/4)\beta}$$

for the constant β in the statement of Theorem 3. This is the case for α sufficiently small.

We also must have

$$r \geq 1/(\beta\delta^2) = \Theta(n^{\alpha/8}/\beta)$$

which is true because $r = \Theta(n^{\alpha/4})$ as observed above.

The output length is $m = \ell^{\Omega(1)} = n^{\Omega(1)}$. The error of the extractor is $2^{-\ell^{\Omega(1)}} = 2^{-n^{\Omega(1)}}$.

Combined with the above error of $2^{-k/2}$ arising from the convex combination, we obtain a total error of again $2^{-n^{\Omega(1)}}$.

For the lower bound for sampling inner product we make use of the following theorem.

Theorem 4 ([8]). *Let X_1 and X_2 be two independent sources on ℓ bits. Suppose the sum of the min-entropies is $\geq \ell + 2\lg(1/\epsilon)$. Then $|\Pr[IP(X_1, X_2) = 1] - 1/2| \leq \epsilon$.*

We now prove our sampling lower bound for inner product.

Proof (of Theorem 2). Suppose there was such a Turing machine. Consider the Turing machine M' that first samples $(X_1, X_2, IP(X_1, X_2))$ then if $IP(X_1, X_2) = 1$ it outputs (X_1, X_2), otherwise it outputs a uniform n-bit string. M' can be implemented, say, in time $O(t)$ with $O(2^q)$ states.

The machine M' samples a distribution (X_1', X_2') with min-entropy $k \geq n-1$. Moreover, because $\Pr[IP(X_1, X_2) = 1]$ approaches $1/2$ for large n, we see that $\Pr[IP(X_1', X_2') = 1]$ approaches $3/4$ for large n.

Set $b := 0.01n$. By Lemma 1, (X_1', X_2') is a convex combination of sources S_j such that except with probability 0.01 over the choice of an independent source from this combination, S_j has min-entropy

$$\begin{aligned}
&\geq n - O(1) - O(q(\lg t)t/b) - \lg(1/0.01) \\
&\geq n - O(n^{\alpha/2}(\lg n)n^{1-\alpha} - O(1) \\
&\geq 0.99n.
\end{aligned}$$

Moreover, each S_j is $S_j = Y_1Y_2$ for independent Y_1, Y_2 and $\ell \leq |Y_1| \leq \ell + b$, where $n = 2\ell + b$. Assume without loss of generality that $|Y_1| \geq |Y_2|$. By conditioning on the $b = 0.01n$ middle bits (each of which depends on exclusively Y_1 or Y_2), we can further write (Y_1, Y_2) as a convex combination of $\leq 2^b$ sources S_j' where each S_j' is $S_j' = Y_1'Y_2'$ where $|Y_1'| = |Y_2'| = n/2$ and Y_1', Y_2' are independent. $Y_1'Y_2'$ has min-entropy $\geq 0.99n - 0.01n = 0.98n$.

This min-entropy is larger than $n/2 + 2\lg(100)$. Hence by Theorem 4 IP will successfully extract one bit with error 0.01.

Overall, the error of the extracted bit is $\leq 0.01 + 0.01 = 0.02$. This contradicts the above remark that $\Pr[IP(X_1', X_2') = 1]$ approaches $3/4$ for large n.

In this proof the extractor is applied to the whole sample, whereas in the proof of Theorem 1 it is applied to a projection of it. That was only for convenience. One could have applied the extractor to the whole sample and then condition on the values of the runs of b bits.

References

1. Aaronson, S.: The equivalence of sampling and searching. In: Computer Science Symp. in Russia (CSR), pp. 1–14 (2011)
2. Ambainis, A., Schulman, L.J., Ta-Shma, A., Vazirani, U.V., Wigderson, A.: The quantum communication complexity of sampling. SIAM J. Comput. 32(6), 1570–1585 (2003)
3. Barak, B., Impagliazzo, R., Wigderson, A.: Extracting randomness using few independent sources. SIAM J. Comput. 36(4), 1095–1118 (2006)
4. Barak, B., Kindler, G., Shaltiel, R., Sudakov, B., Wigderson, A.: Simulating independence: New constructions of condensers, ramsey graphs, dispersers, and extractors. J. of the ACM 57(4) (2010)
5. Barak, B., Rao, A., Shaltiel, R., Wigderson, A.: 2-source dispersers for sub-polynomial entropy and Ramsey graphs beating the Frankl-Wilson construction. In: ACM Symp. on the Theory of Computing (STOC), pp. 671–680 (2006)
6. Beck, C., Impagliazzo, R., Lovett, S.: Large deviation bounds for decision trees and sampling lower bounds for AC0-circuits. Electronic Colloquium on Computational Complexity (ECCC) 19, 42 (2012)
7. Bourgain, J.: More on the sum-product phenomenon in prime fields and its applications. Int. J. of Number Theory (IJNT) 1, 1–32 (2005)
8. Chor, B., Goldreich, O.: Unbiased bits from sources of weak randomness and probabilistic communication complexity. SIAM J. on Computing 17(2), 230–261 (1988)
9. De, A., Watson, T.: Extractors and Lower Bounds for Locally Samplable Sources. In: Goldberg, L.A., Jansen, K., Ravi, R., Rolim, J.D.P. (eds.) APPROX/RANDOM 2011. LNCS, vol. 6845, pp. 483–494. Springer, Heidelberg (2011)
10. Gabizon, A., Raz, R., Shaltiel, R.: Deterministic extractors for bit-fixing sources by obtaining an independent seed. SIAM J. on Computing 36(4), 1072–1094 (2006)
11. Goldreich, O., Goldwasser, S., Nussboim, A.: On the implementation of huge random objects. SIAM J. Comput. 39(7), 2761–2822 (2010)
12. Hennie, F.C.: Crossing sequences and off-line turing machine computations. In: Symposium on Switching Circuit Theory and Logical Design (SWCT) (FOCS), pp. 168–172 (1965)
13. Hopcroft, J.E., Ullman, J.D.: Formal languages and their relation to automata. Addison-Wesley Longman Publishing Co., Inc. (1969)
14. Impagliazzo, R., Nisan, N., Wigderson, A.: Pseudorandomness for network algorithms. In: 26th ACM Symp. on the Theory of Computing (STOC), pp. 356–364 (1994)
15. Jerrum, M.R., Valiant, L.G., Vazirani, V.V.: Random generation of combinatorial structures from a uniform distribution. Theoretical Computer Science 43(2-3), 169–188 (1986)
16. Kamp, J., Rao, A., Vadhan, S.P., Zuckerman, D.: Deterministic extractors for small-space sources. J. Comput. Syst. Sci. 77(1), 191–220 (2011)
17. Kushilevitz, E., Nisan, N.: Communication complexity. Cambridge University Press (1997)
18. Li, X.: Improved constructions of three source extractors. In: IEEE Conf. on Computational Complexity, CCC (2011)
19. Lovett, S., Viola, E.: Bounded-depth circuits cannot sample good codes. Computational Complexity 21(2), 245–266 (2012)
20. Rao, A.: Extractors for low-weight affine sources. In: IEEE Conf. on Computational Complexity (CCC), pp. 95–101 (2009)

21. Raz, R.: Extractors with weak random seeds. In: ACM Symp. on the Theory of Computing (STOC), pp. 11–20 (2005)
22. Santha, M., Vazirani, U.V.: Generating quasi-random sequences from semi-random sources. J. of Computer and System Sciences 33(1), 75–87 (1986)
23. Shaltiel, R.: How to get more mileage from randomness extractors. Random Struct. Algorithms 33(2), 157–186 (2008)
24. Trevisan, L., Vadhan, S.: Extracting randomness from samplable distributions. In: IEEE Symp. on Foundations of Computer Science (FOCS), pp. 32–42 (2000)
25. Viola, E.: Extractors for circuit sources. In: IEEE Symp. on Foundations of Computer Science, FOCS (2011)
26. Viola, E.: The complexity of distributions. SIAM J. on Computing 41(1), 191–218 (2012)