Randomness extractors – applications and constructions

Avi Wigderson
Institute for Advanced Study
Princeton, NJ
avi@ias.edu

ABSTRACT. Randomness extractors are efficient algorithms which convert weak random sources into nearly perfect ones. While such purification of randomness was the original motivation for constructing extractors, these constructions turn out to have strong pseudorandom properties which found applications in diverse areas of computer science and combinatorics. We will highlight some of the applications, as well as recent constructions achieving near-optimal extraction.

Introduction

The quest to purify the randomness in "weak" random sources (of biased and correlated bits) was initiated in the papers of Blum [1] and Santha and Vazirani [16].

The amount of randomness in a distribution for this purpose is captured by the notion of min-entropy, first suggested in this context by Chor and Goldreich [2] and Zuckerman [22]. We say that a random variable has min entropy $\geq k$ if its probability of it hitting any specific value is at most 2^{-k}.

Purifying the randomness from such distributions is captured by the notion of extractors, first defined in the seminal paper of Nisan and Zuckerman [13]. A (k, ϵ)-extractor is a function $E : \{0,1\}^n \times \{0,1\}^d \mapsto \{0,1\}^m$ such that for every random variable X with min entropy k, the distribution of $E(X, U_d)$ has statistical distance $\leq \epsilon$ from the uniform distribution, where U_d denotes a random variable independent of X and uniform on $\{0,1\}^d$. The input U_d is called a *seed* and is thought of as being much shorter (in bits) than X. It is not hard to see that a seed is essential for an extractor to work in this general setting. Such extractors are often called "seeded extractors", to distinguish them from "seedless extractors" (such determinsitic seedless extractors can work only when additional structure is imposed on the source, and will not be discussed here). An excellent survey of seeded extractors is [15].

An extractor has three important parameters. The first is the seed length d, which we wish to minimize. The second is the output length m, which we want to maximize (we want to have $m \approx k$). The third parameter we wish to minimize is the 'error' ϵ – the statistical distance of the output of the extractor from the uniform distribution. It can be shown, using the probabilistic method, that a random function gives an extractor which is optimal in all three parameters, which allows (roughly) $m = k$ and $d = \log(n/\epsilon^2)$. A random function, however, is not satisfactory since in applications we need to be able to compute the extractor efficiently. An extractor which is efficiently computable is called *explicit*. Below we list the progress on explicit constructions, as well as the numerous applications of such explicit extractors.

Foundations of Software Technology and Theoretical Computer Science (Kanpur) 2009.
Editors: Ravi Kannan and K. Narayan Kumar; pp 471–473
Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany.
Digital Object Identifier: 10.4230/LIPIcs.FSTTCS.2009.2340

Constructions Since the 80's there many works devised a variety of techniques to construct explicit extractors of better and better parameters (see [15] for a complete list of references). The first paper to give an explicit extractor which was optimal (up to constant factors) both in seed length and in entropy output was the work of Lu, Reingold, Vadhan and Wigderson [12]. The first to achive this for the error parameter as well were Guruswami, Umans and Vadhan [8], in an elegant construction based on list-decodable Parvaresh-Vardy codes [14], which is also much simpler than [12]. An alternative construction, with the same parameters based on the resolution of the Kakeya conjecture in finite fields [4], was give by Dvir and Wigderson [5]. In all of these the output m was a constant fraction (arbitrarily close to 1) of k. This year Dvir, Kopparty, Saraf and Sudan [6] managed to extract $m = (1 - o(1))k$ for the first time, as byproduct of tight analysis of the Kakeya conjecture. Achieving $m = k$ and removing the large constant factor in the seed length remain challenging openquestions, of relevance to some of the applications.

Applications Extractors posses remarkable pseudorandom properties, which have found applications in a remarkably diverse areas. We list here only some of them, with sample references of each, noting that there are many others.

- Probabilistic algorithms with weak randomness [20, 22, 18]
- Derandomizing small-space computations [13, 10]
- List-decodable error-correcting codes [17]
- Expanders beating the eigenvalue bound (and the applications of these) [21]
- Lossless expanders (and the applications of these) [3]
- Sampling and Hashing [7, 9]
- Cryptography [19]
- Pseudorandom generators [18]
- Metric embeddings [11]

References

[1] Manuel Blum. Independent unbiased coin flips from a correlated biased source: a finite state Markov chain. *Proc. of the 25th FOCS*, 425–433, 1984.

[2] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, April 1988. Special issue on cryptography.

[3] Mike Capalbo, Omer Reingold, Salil Vadhan, and Avi Wigderson. Randomness conductors and constant degree expansion beyond the degree/2 barrier. In *Proceedings of the 34th Annual ACM STOC*, 659–668, 2002.

[4] Zeev Dvir. On the size of Kakeya sets in finite fields. *J. AMS*, 22, 1093–1097, 2009.

[5] Zeev Dvir, Avi Wigderson. Kakeya sets, new mergers and old extractors. *Proc. of the 49th FOCS*, 625–633, 2008.

[6] Zeev Dvir, Swastik Kopparty, Shubhangi Saraf and Madhu Sudan. Extensions to the method of multiplicity, with applications to Kakeya sets and mergers. proc. of FOCS '09, 2009, to appear.

[7] Oded Goldreich. A sample of samplers. *ECCC*, TR97-020, 1997.

[8] Venkat Guruswami, Chris Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from parvaresh-vardy codes. In *Proceedings of the 22nd Conference on Computational Complexity*, pages 96–108, 2007.

[9] Oded Goldreich and Avi Wigderson. Tiny families of functions with random properties: A quality-size trade-off for hashing. *Random Structures & Algorithms*, 11(4):315–343, 1997.

[10] Oded Goldreich and Avi Wigderson. Derandomization that is rarely wrong from short advice that is typically good. *Proc. of the 6th RANDOM conference*, 209–223, 2002.

[11] Pyotr Indik. Uncertainty principles, extractors and explicit embeddings of L2 into L1. *Proc. of the 39th STOC*, 2007.

[12] 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"*, 602–611, 2003.

[13] Noam Nisan and David Zuckerman. Randomness is Linear in Space. *JCSS*, 43–52, 1996.

[14] Farzad Parvaresh and Alexander Vardy. Correcting errors beyond the guruswami-sudan radius in polynomial time. In *Proceedings of the 46th FOCS*, pages 285–294, Washington, DC, USA, 2005. IEEE Computer Society.

[15] Ronen Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77:67–95, 2002.

[16] Miklos Santha and Umesh Vazirani. Generating quasi-random sequences from semi-random sources. *JCSS*, 48(4), 860–879, 2001

[17] Amnon Ta-Shma and David Zuckerman. Extractor codes. *Proc. of the 33rd STOC*, 193–199, 2001.

[18] Luca Trevisan. Extractors and pseudorandom generators. *JACM*, 33, 75–87, 1986.

[19] Salil Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *Cryptology*, 17(1), 43–77, 2004.

[20] Umesh Vazirani and Vijay vazirani. Random polynomial time is equal to semi-random polynomial time. *Proc. 26th FOCS*, 417–428, 1985.

[21] Avi Wigderson and David Zuckerman. Expanders that beat the eigenvalue abound: explicit construction and applications. *Combinatorica*, 19(1), 125–138, 1999.

[22] David Zuckerman. Simulating BPP using a weak random source. *Algorithmica*, 16(4/5), 1996.