The Complexity of Estimating Min-Entropy

Thomas Watson*

April 14, 2014

Abstract

Goldreich, Sahai, and Vadhan (CRYPTO 1999) proved that the promise problem for estimating the Shannon entropy of a distribution sampled by a given circuit is NISZK-complete. We consider the analogous problem for estimating the *min-entropy* and prove that it is SBP-complete, where SBP is the class of promise problems that correspond to approximate counting of NP witnesses. The result holds even when the sampling circuits are restricted to be 3-local. For logarithmic-space samplers, we observe that this problem is NP-complete by a result of Lyngsø and Pedersen on hidden Markov models (JCSS 2002).

1 Introduction

Deterministic randomness extraction is the problem of taking a sample from an imperfect physical source of randomness (modeled as a probability distribution on bit strings) and applying an efficient deterministic algorithm to transform it into a uniformly random string, which can be used by a randomized algorithm (see [Sha11, Vad12] for surveys of this topic). For such extraction to be possible, the source of randomness must satisfy two properties: (i) it must contain a sufficient "amount of randomness", and (ii) it must be "structured", meaning that it has a simple description.

Regarding property (i), the most useful measure of the "amount of randomness" is the *min-entropy*, which is the logarithm of the reciprocal of the probability of the most likely outcome (in other words, if a distribution has high min-entropy then every outcome has small probability). This proposal originates in [CG88]. The number of uniformly random bits produced by the extractor cannot exceed the min-entropy of the source, and one of the goals in designing extractors is to get as close to the min-entropy as possible. Regarding property (ii), if the distribution is generated by an efficient process in the physical world, then it can be modeled as being sampled by an efficient algorithm given uniform random bits. This sampling algorithm is a simple description of the distribution. Trevisan and Vadhan [TV00] initiated the study of extracting from efficiently samplable distributions. Assuming certain complexity-theoretic conjectures, they constructed extractors for time-efficient samplers.[1] Kamp et al. [KRVZ11] gave an *unconditional* construction of extractors for space-efficient samplers with streaming (one-way) access to their random input bits. De and Watson [DW12] and Viola [Vio11] gave constructions of extractors for *local* samplers (where each

*Department of Computer Science, University of Toronto. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-0946797 and by the National Science Foundation under Grant No. CCF-1017403.

[1] We mention that a somewhat related but incomparable problem was studied in [DRV12].

output bit of the sampler only depends on a small number of the random input bits), and Viola [Vio11] generalized this to get extractors for samplers that are constant-depth circuits (of the AC^0 type). Viola [Vio12] has constructed extractors for sequential-access one-tape Turing machine samplers.

All of these extractor constructions need to be given a lower bound on the min-entropy of the distribution. The output length of the extractor depends on this lower bound. Thus, if we had a sampling algorithm (assumed to model the physical source), it would be nice to know the min-entropy so we could plug this parameter into the extractor, and thus extract as much of the randomness as possible.[2] This motivates the following computational problem: Given an efficient algorithm that outputs a sample from a probability distribution, estimate the min-entropy of the distribution. The upshot of our results is that this problem is intractable even for the extremely simple samplers studied in [KRVZ11, DW12, Vio11], and we pinpoint the precise complexity of the problem.

Goldreich, Sahai, and Vadhan [GSV99] considered the problem of estimating the *Shannon entropy* of a distribution sampled by a given circuit. They showed that an appropriate formulation of the problem is complete for the complexity class NISZK (non-interactive statistical zero-knowledge) and is thus believed to be intractable. For the *min-entropy* version, we show that the problem is interreducible with the "approximate lower bound" problem that was famously studied by Goldwasser and Sipser [GS86]. The latter formulation of multiplicative approximate counting of NP witnesses deserves its own complexity class. Indeed, the class has already been named SBP by [BGM06], and it is perhaps the only natural example of a class sandwiched between MA and AM. We prove that the min-entropy estimation promise problem is SBP-complete even when restricted to 3-local samplers (as studied in [DW12, Vio11]).

For logarithmic-space samplers that have one-way access to their randomness (as studied in [KRVZ11]), it turns out that our min-entropy estimation promise problem has already been studied (though in a very different context and with different terminology) by Lyngsø and Pedersen [LP02], who proved that an equivalent problem is NP-complete. We discuss the relationship between their problem and our problem.

1.1 Definitions

The *min-entropy* of a distribution D over a finite set S is $\mathrm{H}_\infty(D) = \min_{s \in S} \log_2\big(1/\Pr_D[s]\big)$, where $\Pr_D[s]$ denotes the probability assigned to outcome s by D. Let U_r denote the uniform distribution over $\{0,1\}^r$. If $A : \{0,1\}^r \to \{0,1\}^m$ is an algorithm that takes r uniformly random bits and outputs m bits, then $A(U_r)$ denotes the output distribution of A. We write $A(U)$ with the convention that $U = U_r$ for the appropriate value of r. We consider three classes of sampling algorithms.

- *Circuits*, i.e., the usual boolean circuits.
- *d-Local* samplers are functions where each of the m output bits depends on at most d of the r input bits (where d is a constant).
- *Logarithmic-space* samplers can be defined in several equivalent ways; the following is the most convenient for us. The sampler is a layered directed graph where each edge goes from

[2] We remark that the aforementioned extractor constructions *do not* assume knowledge of the sampling algorithm itself, only knowledge of the class of algorithms the sampler comes from. It is not known how to exploit knowledge of the description of the distribution for extraction purposes, except in trivial cases such as when the distribution is uniform over an affine subspace of $GF(2)^n$.

one layer to the immediate next layer. There is a unique start vertex in layer 0. For each vertex except the ones in the last layer, there is at least one outgoing edge, and the outgoing edges are labeled with a probability distribution. Each vertex except the start vertex is labeled with a bit. A sample is obtained by taking a random walk (starting at the start vertex) and outputting the bit labels of the visited vertices.[3] Every logarithmic space Turing machine (with a hard-wired input and with one-way access to randomness) can be modeled as such a sampler (of polynomial size) by letting the vertices in layer i represent all the possible configurations the machine could be in right after the i^{th} output bit has been produced.

Such d-local samplers and logarithmic-space samplers have been studied in other contexts besides randomness extraction. For example, there are many positive and negative results on whether d-local samplers can implement pseudorandom generators and one-way functions (see [App14] for a survey). Trevisan et al. [TVZ05] showed how to efficiently perform near-optimal prefix-free compression of distributions with logarithmic-space samplers.

The min-entropy estimation problem that we study is formulated in terms of promise problems (see [Gol06] for a survey on promise problems).

Definition 1. *For any class $\mathcal{A}$ of algorithms, $\mathcal{A}$-*Min-Ent-Gap *is the following promise problem.*

$$\begin{aligned} \mathcal{A}\text{-}\textsc{Min-Ent-Gap}_{\text{YES}} &= \big\{(A,h) \ : \ A \in \mathcal{A} \text{ and } \mathrm{H}_\infty(A(U)) \le h\big\} \\ \mathcal{A}\text{-}\textsc{Min-Ent-Gap}_{\text{NO}} &= \big\{(A,h) \ : \ A \in \mathcal{A} \text{ and } \mathrm{H}_\infty(A(U)) > h+1\big\} \end{aligned}$$

Taking $\mathcal{A}$ to be circuits, d-local samplers, or logarithmic-space samplers, we get the problems Circuit-Min-Ent-Gap, d-Local-Min-Ent-Gap, and Logspace-Min-Ent-Gap. The size of the input (A,h) is the bit length of the description of the algorithm A, plus the bit length of the integer h. Note that if one of these problems has a polynomial-time algorithm, then the min-entropy can be estimated within an additive 1 in polynomial time by trying all possible values of $h \in \{0,1,\ldots,m\}$ (or more efficiently by using binary search).

Throughout this paper, when we talk about reductions and completeness, we are always referring to deterministic polynomial-time mapping reductions.

Definition 2. prSBP *is the class of promise problems $\Pi = (\Pi_{\text{YES}}, \Pi_{\text{NO}})$ for which there exist polynomial-time algorithms M and K (where M outputs a bit and K outputs a nonnegative integer) and a polynomial p such that the following hold for all $x \in \{0,1\}^*$.*

$$\begin{aligned} x \in \Pi_{\text{YES}} &\implies \big|\big\{y \in \{0,1\}^{p(|x|)} \ : \ M(x,y)=1\big\}\big| \ge K(x) \\ x \in \Pi_{\text{NO}} &\implies \big|\big\{y \in \{0,1\}^{p(|x|)} \ : \ M(x,y)=1\big\}\big| < K(x)/2 \end{aligned}$$

SBP *is defined as the class of sets in* prSBP*.*

The factor of $1/2$ in the gap in Definition 2 is arbitrary and can be replaced by $1 - 1/q(|x|)$ for any polynomial q, by a standard trick.[4]

[3] The model in [KRVZ11] is the same except the output bits are on the edges rather than on the vertices (Mealy style rather than Moore style). The two models are equivalent up to a small difference in the size of the graph.

[4] Modify K so its output is raised to the power q, and modify M so its number of accepted strings y is also raised to the power q (by taking $y_1, \ldots, y_{q(|x|)} \in \{0,1\}^{p(|x|)}$ and accepting iff $M(x,y_i)=1$ for all i).

Observation 3. prSBP *is the class of all promise problems reducible to the following promise problem* Circuit-Count-Gap.

$$\begin{aligned}\textsc{Circuit-Count-Gap}_{\text{YES}} &= \big\{(C,k) \;:\; C \textit{ is a circuit that accepts} \geq k \textit{ inputs}\big\}\\ \textsc{Circuit-Count-Gap}_{\text{NO}} &= \big\{(C,k) \;:\; C \textit{ is a circuit that accepts} < k/2 \textit{ inputs}\big\}\end{aligned}$$

Proof. To reduce an arbitrary $\Pi \in$ prSBP (with associated M, K, and p) to Circuit-Count-Gap, map x to (C_x, k_x) where $C_x(y) = M(x,y)$ and $k_x = K(x)$. Conversely, suppose some Π reduces to Circuit-Count-Gap with x mapping to (C_x, k_x), and note that by adding dummy input bits to C_x we may assume the number of input bits is a polynomial p depending only on $|x|$, not on x. Then $\Pi \in$ prSBP is witnessed by M and K such that $M(x,y) = C_x(y)$ and $K(x) = k_x$. □

Böhler, Glaßer, and Meister [BGM06] introduced the class SBP (which stands for "small bounded-error probability") and provided a fairly comprehensive study of it from a structural complexity perspective, analyzing its relationship to other classes (inclusions and relativized separations), its closure properties, and the possibility of it having complete sets. Note that MA $\subseteq$ SBP $\subseteq$ AM, where MA $\subseteq$ SBP follows by observing that the standard proof of MA $\subseteq$ PP [Ver92] automatically yields a multiplicative gap, and SBP $\subseteq$ AM follows immediately from the Goldwasser-Sipser lower bound protocol [GS86]. Both inclusions relativize. There is an oracle relative to which SBP $\not\subseteq \Sigma_2$P [San89, BGM06] and thus MA $\neq$ SBP (since MA $\subseteq \Sigma_2$P relativizes), and there is an oracle relative to which AM $\not\subseteq$ PP [Ver92] and thus SBP $\neq$ AM (since SBP $\subseteq$ PP relativizes). Since AM can be derandomized to NP under reasonable complexity assumptions [KvM02, AK01, MV05, SU06], it is believed that SBP = NP.

Although very few papers explicitly mention the class SBP, the Goldwasser-Sipser protocol for Circuit-Count-Gap has many applications in complexity and cryptography, and thus SBP has been implicitly studied many times. For example, it is shown in [AAB+10, AGHK11] that E^{prSBP} contains sets of circuit complexity $\Omega(2^n/n)$.

1.2 Results

Theorem 4. Circuit-Min-Ent-Gap *and* 3-Local-Min-Ent-Gap *are both* prSBP*-complete.*

Theorem 5. Logspace-Min-Ent-Gap *is* prNP*-complete.*

We prove Theorem 4 in Section 2. Our proof shows that the completeness holds even when each output bit of the sampler is the disjunction of exactly three unnegated input bits. Note that the min-entropy of a 1-local sampler's distribution is trivial to compute exactly since the distribution is affine. The complexity of estimating min-entropy for 2-local samplers remains open. During our proof of Theorem 4, we also show that Monotone-2-Sat-Count-Gap is prSBP-complete.[5] It was previously known (presumably folklore) that this problem is in prBPP iff NP = RP (roughly speaking, it is "NP-complete modulo randomness"). This is implied by our result, which more precisely quantifies the complexity in terms of deterministic reductions.

[5] Monotone-2-Sat-Count-Gap is defined in the natural way, as the restriction of Circuit-Count-Gap to instances where C is a 2-Sat formula with no negated literals.

We discuss Theorem 5 in Section 3. The prNP-hardness follows without difficulty from a result of Lyngsø and Pedersen [LP02] on hidden Markov models.[6] To cut to the chase, the only issue is that their result allows the sampler to output strings of different lengths, while our definition requires it to output fixed-length strings. This issue is rather straightforward to resolve.

In the proof of Theorem 4, we implicitly use a "closure under nondeterminism" property of SBP (specifically, in Claim 6), which was not shown in [BGM06]. This is analogous to how AM (and trivially, MA) is "closed under nondeterminism". In Section 4 we explicitly state and prove a more general form of this property of SBP. The general form is not needed for our theorems about min-entropy, but it demonstrates the robustness of the class SBP and may be useful for future study of SBP.

1.3 Related Work

Goldreich et al. [GSV99] showed that the variant of Circuit-Min-Ent-Gap where Shannon entropy replaces min-entropy and the roles of YES and NO instances are interchanged is prNISZK-complete. Dvir et al. [DGRV11] gave some upper and lower bounds on the complexity of estimating various types of entropy for distributions where a sample is obtained by plugging a uniform input into a sequence of low-degree multivariate polynomials over a finite field. Dvir et al. [DGRV11] also studied the complexity of estimating Shannon entropy for d-local samplers and for logarithmic-space samplers that have *two*-way access to their randomness.

Other papers that are in a somewhat similar spirit as ours include [MU02, FIKU08, BMV08, BBM11]. We refer to [DW12] for an overview of past work on locally computable functions.

The study of multiplicative approximate counting of NP witnesses was initiated in [Sto85]. Derandomization of approximate counting was studied in [SU06]. See [DGGJ03] for a more algorithmic perspective on the complexity of approximate counting. Kuperberg [Kup09] showed that two variants of SBP are actually equal: SBQP (the quantum variant) and $\mathrm{A_0PP}$ (the variant where we consider a difference of two #P functions rather than a single #P function). Kabanets et al. [KRC00] defined and studied a complexity class that captures *additive* approximate counting in a more direct way than BPP does.

2 Proof of Theorem 4

Claim 6. Circuit-Min-Ent-Gap $\in$ prSBP.

Claim 7. Circuit-Min-Ent-Gap *is* prSBP-*hard.*

Claim 8. d-Sat-Count-Gap *reduces to* $(d+1)$-Local-Min-Ent-Gap.

Claim 9. Monotone-2-Sat-Count-Gap *is* prSBP-*hard.*

Claim 6 and Claim 7 constitute the prSBP-completeness of Circuit-Min-Ent-Gap. To see the prSBP-completeness of 3-Local-Min-Ent-Gap, note that it is in prSBP since it trivially reduces

[6] Interestingly, it is also shown in [LP02] that for logarithmic-space samplers, estimating the statistical distance between two distributions is closely related to estimating the *min-entropy* of a distribution. In contrast, for polynomial-size circuits it is known that estimating the statistical distance is closely related to estimating the *Shannon entropy* (see [Vad13]).

to Circuit-Min-Ent-Gap (which is in prSBP by Claim 6), and combining Claim 9 with Claim 8 using $d = 2$ implies that 3-Local-Min-Ent-Gap is prSBP-hard. Of course, Claim 7 is implied by Claim 8 and Claim 9; however, the proof of Claim 7 serves as a warmup for the proof of Claim 8, and only the former is needed for showing that Circuit-Min-Ent-Gap is prSBP-complete.

By Claim 9 (and Observation 3), Monotone-2-Sat-Count-Gap is prSBP-complete. This problem was previously known to be prNP-hard, by combining any reduction to Vertex-Cover with a "blow-up" trick that is usually attributed to [JVV86, Sin93]. To improve the prNP-hardness to prSBP-hardness, we need to use a particular reduction to Vertex-Cover, satisfying certain properties.

We now prove the above four claims.

Proof of Claim 6. We reduce Circuit-Min-Ent-Gap to Circuit-Count-Gap. The intuition is as follows. Given a sampling circuit, if we knew which outcome was most probable (under the output distribution), then estimating the min-entropy would be equivalent to estimating the number of preimages of that outcome (which form an NP witness set). Handling the fact that we do not know the most probable outcome involves summing over all outcomes, which necessitates a step to amplify the gap between YES instances and NO instances.

We proceed with the formal proof. Given an instance (A, h) of Circuit-Min-Ent-Gap where $A : \{0,1\}^r \to \{0,1\}^m$ is a circuit and without loss of generality $h \le \min(r, m)$, we construct a circuit $C : \{0,1\}^m \times (\{0,1\}^r)^{m+1} \to \{0,1\}$ by

$$C(x, y_1, \ldots, y_{m+1}) \;=\; \begin{cases} 1 & \text{if } A(y_i) = x \text{ for all } i \\ 0 & \text{otherwise} \end{cases}$$

and let $k = 2^{(r-h)(m+1)}$. We show the following two things.

$$\begin{aligned} (A, h) \in \textsc{Circuit-Min-Ent-Gap}_{\text{YES}} &\implies (C, k) \in \textsc{Circuit-Count-Gap}_{\text{YES}} \\ (A, h) \in \textsc{Circuit-Min-Ent-Gap}_{\text{NO}} &\implies (C, k) \in \textsc{Circuit-Count-Gap}_{\text{NO}} \end{aligned}$$

For the YES case, the assumption $\mathrm{H}_\infty(A(U_r)) \le h$ means there exists an $x \in \{0,1\}^m$ such that $\Pr_{A(U_r)}[x] \ge 1/2^h$ and thus there are at least 2^{r-h} strings y for which $A(y) = x$. Thus there are at least $2^{(r-h)(m+1)}$ choices of $y_1, \ldots, y_{m+1}$ for which $A(y_i) = x$ for all i, which implies that C accepts at least k inputs. For the NO case, the assumption $\mathrm{H}_\infty(A(U_r)) > h + 1$ means that for all $x \in \{0,1\}^m$, $\Pr_{A(U_r)}[x] < 1/2^{h+1}$ and thus there are less than 2^{r-h-1} strings y for which $A(y) = x$. Thus there are less than $2^{(r-h-1)(m+1)}$ choices of $y_1, \ldots, y_{m+1}$ for which $A(y_i) = x$ for all i. By summing over x, this implies that C accepts less than $2^m \cdot 2^{(r-h-1)(m+1)} = k/2$ inputs. □

Proof of Claim 7. We reduce Circuit-Count-Gap to Circuit-Min-Ent-Gap. The intuition is as follows. Given a circuit, we would like to transform it into a sampler such that satisfying assignments of the circuit somehow correspond to preimages of the most probable outcome of the sampler (so that counting these two things are equivalent). Using the circuit itself as the sampler has the issue that 1 may not be the most probable outcome, so we fix this by replacing the outcome 0 with a very high-min-entropy distribution (so it does not affect the overall min-entropy much).

We proceed with the formal proof. Given an instance (C, k) of Circuit-Count-Gap where $C : \{0,1\}^n \to \{0,1\}$ is a circuit and without loss of generality $1 \le k \le 2^n$, by standard amplification we may assume that C accepts at least k inputs in the YES case and less than $k/8$ inputs in the NO case. We construct a circuit $A : \{0,1\}^n \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ by

$$A(y,z) \;=\; \begin{cases} 1^{2n} & \text{if } C(y) = 1 \\ z & \text{otherwise} \end{cases}$$

and let h be the smallest integer such that $1/2^h \leq k/2^n$. We show the following two things.

$$\begin{aligned} (C,k) \in \textsc{Circuit-Count-Gap}_{\text{YES}} &\implies (A,h) \in \textsc{Circuit-Min-Ent-Gap}_{\text{YES}} \\ (C,k) \in \textsc{Circuit-Count-Gap}_{\text{NO}} &\implies (A,h) \in \textsc{Circuit-Min-Ent-Gap}_{\text{NO}} \end{aligned}$$

For the YES case, the assumption that C accepts at least k inputs implies that $\Pr_{A(U_{3n})}[1^{2n}] \geq k/2^n \geq 1/2^h$ and thus $\mathrm{H}_\infty(A(U_{3n})) \leq h$. For the NO case, the assumption that C accepts less than $k/8$ inputs implies that

$$\Pr_{A(U_{3n})}[1^{2n}] \;\leq\; \Pr_{y \sim U_n}[C(y)=1] + \Pr_{z \sim U_{2n}}[z = 1^{2n}] \;<\; (k/8)/2^n + 1/2^{2n} \;<\; (k/4)/2^n \;<\; 1/2^{h+1}$$

by the minimality of h. Since 1^{2n} is the most probable string under $A(U_{3n})$, this implies that $\mathrm{H}_\infty(A(U_{3n})) > h+1$. □

Proof of Claim 8. The intuition is similar to the intuition for Claim 7, except that now we need to ensure the sampler is local. Checking whether the output of a d-Sat formula is 0 cannot be done locally (since it depends on all variables). Instead, observe that it suffices to check for each clause separately whether it is satisfied and, if it is not, then cause the output to have very high min-entropy (conditioned on the unsatisfying assignment). This can be performed in a local manner.

We proceed with the formal proof. Given an instance (φ,k) of d-Sat-Count-Gap,[7] where φ is a d-Sat formula having n variables and m clauses and without loss of generality $1 \leq k \leq 2^n$, by standard amplification we may assume that φ has at least k satisfying assignments in the YES case and less than $k/8$ satisfying assignments in the NO case. (This does not affect the constant d, since the amplified formula is just a conjunction of several copies of the original d-Sat formula on disjoint variables.) Suppose the i^{th} clause of φ consists of the literals $\ell_{i,1} \vee \cdots \vee \ell_{i,d}$. We construct a $(d+1)$-local function $A : \{0,1\}^n \times \{0,1\}^{2n} \to \{0,1\}^{m \cdot 2n}$ where the first input is the variables of φ (denoted y) and the bits of the second input are labeled as z_j. We let the (i,j) bit of the output (for $i \in \{1,\ldots,m\}$, $j \in \{1,\ldots,2n\}$) be

$$A(y,z)_{i,j} \;=\; \ell_{i,1} \vee \cdots \vee \ell_{i,d} \vee z_j$$

and let h be the smallest integer such that $1/2^h \leq k/2^n$. We show the following two things.

$$\begin{aligned} (\varphi,k) \in d\text{-}\textsc{Sat-Count-Gap}_{\text{YES}} &\implies (A,h) \in (d+1)\text{-}\textsc{Local-Min-Ent-Gap}_{\text{YES}} \\ (\varphi,k) \in d\text{-}\textsc{Sat-Count-Gap}_{\text{NO}} &\implies (A,h) \in (d+1)\text{-}\textsc{Local-Min-Ent-Gap}_{\text{NO}} \end{aligned}$$

Note that for any fixed assignment y, if the i^{th} clause is satisfied then for all j the (i,j) output bit is 1 (with probability 1 over random z), and if the i^{th} clause is not satisfied then the (i,j) output bits are uniformly distributed (equal to z). It follows that if y satisfies φ then $\Pr_{A(y,U_{2n})}[1^{m \cdot 2n}] = 1$, and that if y does not satisfy φ then $\Pr_{A(y,U_{2n})}[1^{m \cdot 2n}] = 1/2^{2n}$ and for all $s \in \{0,1\}^{m \cdot 2n}$, $\Pr_{A(y,U_{2n})}[s] \in$

[7] In fact, the proof works for arbitrary d-CSPs over the binary alphabet.

$\{0, 1/2^{2n}\}$. In either case, $1^{m\cdot 2n}$ is a most probable string under $A(y, U_{2n})$, and thus $1^{m\cdot 2n}$ is a most probable string under $A(U_{3n})$.

For the YES case, the assumption that φ has at least k satisfying assignments implies that

$$\Pr_{A(U_{3n})}[1^{m\cdot 2n}] \;\geq\; \Pr_{y\sim U_n}[y \text{ satisfies } \varphi] \;\geq\; k/2^n \;\geq\; 1/2^h$$

and thus $\mathrm{H}_\infty(A(U_{3n})) \leq h$. For the NO case, the assumption that φ has less than $k/8$ satisfying assignments implies that

$$\begin{aligned}\Pr_{A(U_{3n})}[1^{m\cdot 2n}] &= 1\cdot \Pr_{y\sim U_n}[y \text{ satisfies } \varphi] + (1/2^{2n})\cdot \Pr_{y\sim U_n}[y \text{ does not satisfy } \varphi]\\ &< (k/8)/2^n + 1/2^{2n}\\ &< (k/4)/2^n\\ &< 1/2^{h+1}\end{aligned}$$

by the minimality of h. Since $1^{m\cdot 2n}$ is a most probable string, this implies that $\mathrm{H}_\infty(A(U_{3n})) > h+1$. □

Proof of Claim 9. We reduce Circuit-Count-Gap to Monotone-2-Sat-Count-Gap. The intuition is as follows. Satisfying assignments of a Monotone-2-Sat formula can be viewed as vertex covers of an associated graph (whose vertices correspond to variables and whose edges correspond to clauses). Hence we would like to reduce from approximately counting satisfying assignments of a circuit to approximately counting vertex covers of a graph. It is straightforward to reduce to approximately counting *small* vertex covers, by using a parsimonious NP-completeness reduction. The "blow-up" trick modifies a graph to ensure there are far more small vertex covers than large vertex covers. However, for this to make approximately counting small vertex covers equivalent to approximately counting vertex covers, we need the parsimonious reduction to ensure that all the "small" vertex covers have the same size. The so-called FGLSS reduction [FGL+96] has the necessary properties.

We proceed with the formal proof. Given an instance (C, k) of Circuit-Count-Gap where without loss of generality $k \geq 1$, by standard amplification we may assume that C accepts at least k inputs in the YES case and less than $k/4$ inputs in the NO case. We first apply the standard parsimonious reduction from Circuit-Sat to 3-Sat to obtain a formula φ with the same number of satisfying assignments as C. Next, we apply the FGLSS reduction from 3-Sat to Vertex-Cover: For each clause of φ, create seven vertices representing the satisfying assignments for the three variables in the clause, and put an edge between two vertices if they conflict (i.e., they assign some variable opposite truth values). Let G denote this graph, and let $\ell = 6m$ where m is the number of clauses in φ. This particular reduction has the following two properties: (i) it is parsimonious (i.e., the number of vertex covers of G of size at most ℓ equals the number of satisfying assignments of φ), and (ii) every vertex cover of G has size at least ℓ.

Next, we apply blow-up to G to create a new graph G', by transforming each vertex of G into a cloud of $10m$ vertices and transforming each edge of G into a complete bipartite graph between its two clouds. We view G' as a Monotone-2-Sat formula φ' where vertices become variables and edges become monotone clauses, and we let $k' = k\cdot(2^{10m}-1)^m$. We show the following two things.

$$\begin{aligned}(C,k) \in \textsc{Circuit-Count-Gap}_{\mathrm{YES}} &\implies (\varphi', k') \in \textsc{Monotone-2-Sat-Count-Gap}_{\mathrm{YES}}\\ (C,k) \in \textsc{Circuit-Count-Gap}_{\mathrm{NO}} &\implies (\varphi', k') \in \textsc{Monotone-2-Sat-Count-Gap}_{\mathrm{NO}}\end{aligned}$$

We first observe that each vertex cover of G, say S of size s, gives rise to $(2^{10m}-1)^{7m-s}$ vertex covers of G'. This is shown as follows: For each cloud representing a vertex in S, include all vertices of the cloud, and for each cloud representing a vertex not in S, include any subset of the cloud except the entire cloud. These are indeed vertex covers of G', and every vertex cover of G' can be obtained uniquely in this way.[8] Stated another way: Partition the set of all vertex covers of G' according to which clouds have all vertices included. Each part of this partition is associated with the set S of vertices in G whose clouds have all their vertices included in those vertex covers of G'. The set S is a vertex cover of G, and if S has size s then the corresponding part of the partition consists of all sets of vertices of G' obtained by choosing, for each of the $7m-s$ vertices of G not in S, one of the $2^{10m}-1$ proper subsets of the associated cloud. Hence the vertex covers of G' are partitioned according to the vertex cover of G they correspond to. The total number of vertex covers of G', and hence the total number of satisfying assignments of φ', is thus

$$\sum_s (2^{10m}-1)^{7m-s} \cdot (\text{number of vertex covers of } G \text{ of size } s).$$

For the YES case, the assumption that C accepts at least k inputs implies that G has at least k vertex covers of size at most ℓ (by Property (i)), which implies that the number of satisfying assignments of φ' is at least $k \cdot (2^{10m}-1)^{7m-\ell} = k'$. For the NO case, the assumption that C accepts less than $k/4$ inputs implies that G has: 0 vertex covers of size less than ℓ (by Property (ii)), less than $k/4$ vertex covers of size exactly ℓ (by Property (i)), and trivially at most 2^{7m} vertex covers of size greater than ℓ. Thus the number of satisfying assignments of φ' is less than

$$(k/4)\cdot(2^{10m}-1)^{7m-\ell} + 2^{7m}\cdot(2^{10m}-1)^{7m-\ell-1} \;=\; k'/4 + k'\cdot 2^{7m}/k(2^{10m}-1) \;<\; k'/2. \qquad \square$$

3 Proof of Theorem 5

First note that Logspace-Min-Ent-Gap $\in$ prNP since the most probable output string can be nondeterministically guessed, and then the probability of that string can be computed exactly by simple dynamic programming (the forward part of the classic forward-backward algorithm). The prNP-hardness follows without difficulty from a result of Lyngsø and Pedersen on hidden Markov models [LP02]; we now elaborate.

A hidden Markov model consists of a (time-invariant) Markov chain with a designated start state, where each state is either "silent" or has a distribution over symbols from some alphabet. When the Markov chain reaches a particular state, it first outputs a symbol from the distribution associated with that state (assuming the state is not silent) and then transitions to another state according to the probabilities on the outgoing edges. Running the Markov chain for a certain number of steps thus yields a random output string whose length is the number of non-silent states visited. The result of [LP02] shows, by a reduction from Max-Clique, that it is NP-hard to estimate the probability of the most likely output string, even in the special case where the Markov chain is a DAG with a unique source and sink (which are the only silent states) and where each non-silent state deterministically outputs a bit. In fact, the result shows that the gap version of

[8] The reason we disallow including the entire cloud corresponding to $v \notin S$ is because if it were included, then the resulting vertex cover of G' would also arise from the vertex cover $S \cup \{v\}$ of G and hence would not be uniquely obtainable.

the estimation problem is prNP-hard with a multiplicative gap of $n^{\Omega(1)}$ (where n is the size of the Markov chain), by using the NP-hardness of approximating Max-Clique (see [Hås99] and the references within).

The issue, however, is that the hidden Markov model produced by the reduction in [LP02] may output bit strings of different lengths on different runs. To summarize, our definition of logarithmic-space samplers and the specific model used in the construction of [LP02] are almost exactly the same: In both cases there is a DAG with a unique source (of indegree 0), and each vertex has a probability distribution on its outgoing edges, and each vertex is either silent or labeled with a bit. A sample is obtained by taking a random walk starting at the source and halting when a sink (of outdegree 0) is reached, and outputting the bit labels of the visited vertices. There are two differences between the two models.

(1) In the construction of [LP02] there is a unique sink, and the source and sink are the only silent vertices. In our model, the source is the only silent vertex, and we do not limit the number of sinks. The sink in the construction of [LP02] can be removed without affecting the output distribution, but it is a technical convenience to leave it in.

(2) The important difference is that our definition further requires that the DAG is layered, and that all sinks are in the last layer (which ensures all output strings have the same length).

Let G denote the output of the reduction of [LP02]. We transform G into a G' that conforms to our definition and whose output distribution is "the same" as that of G, in the following sense. Let m denote the length of a longest path in G, let D denote the output distribution of G (so D is over $\{0,1\}^{\leq m}$, i.e., strings of length at most m), and let D' denote the output distribution of (the to-be-constructed) G'. Consider the following injection $I : \{0,1\}^{\leq m} \to \{0,1\}^{2m}$: Given a string of length at most m, insert a 1 before each bit, and then pad it with 00's until it has length exactly $2m$. We construct G' so that $D' = I(D)$ (i.e., it behaves as if G were run and then I were applied to the output). Then $\mathrm{H}_\infty(D) = \mathrm{H}_\infty(D')$, and this will complete the reduction to Logspace-Min-Ent-Gap. In fact, this shows that it is prNP-hard to estimate the min-entropy with an additive gap of $\Omega(\log n)$ (which is stronger than the gap of 1 stated in Theorem 5).

We construct G' as follows. We first construct an intermediate G'' that has output distribution $D'' = I(D)$ and that conforms to our definition, except that each non-source vertex of G'' outputs *two* bits. Then transforming G'' into G' (where each non-source vertex outputs one bit) is straightforward, by doubling the number of layers. We construct G'' as follows. Take m copies of G (except the source) and put each copy in a separate layer, which represents a time step. Retain all the original transitions but make them go between adjacent layers, and include a source transitioning to the first layer. Letting t denote the sink of G, make each copy of t (except the one in the last layer) always transition to itself in the next layer. Iteratively remove any spurious sources. Make each copy of t output 00, and make the copies of other non-source vertices output 1 followed by their original bit.

4 SBP is Closed Under Nondeterminism

Consider the following nondeterministic generalization of Circuit-Count-Gap: Given (C, k) where C is a circuit that takes two inputs w and y, does there exist a w for which C accepts at least k strings y, or is it the case that for all w, C accepts less than $k/2$ strings y? In showing that Circuit-Min-Ent-Gap $\in$ prSBP, we implicitly showed that this nondeterministic generalization is in prSBP (see Claim 6). We now observe that it remains in prSBP *even if we allow the location of the gap to depend on the nondeterministic guess.* (This is reflected in the following definition by allowing K to be a function of both x and w, rather than merely a function of x.)

Definition 10. prNSBP *is the class of promise problems* $\Pi = (\Pi_{\text{YES}}, \Pi_{\text{NO}})$ *for which there exist polynomial-time algorithms* M *and* K *(where* M *outputs a bit and* K *outputs a nonnegative integer) and a polynomial* p *such that the following hold for all* $x \in \{0,1\}^*$.

$$\begin{aligned} x \in \Pi_{\text{YES}} &\implies \exists w \in \{0,1\}^{p(|x|)} \;:\; \left|\left\{y \in \{0,1\}^{p(|x|)} \;:\; M(x,w,y) = 1\right\}\right| \geq K(x,w) \\ x \in \Pi_{\text{NO}} &\implies \forall w \in \{0,1\}^{p(|x|)} \;:\; \left|\left\{y \in \{0,1\}^{p(|x|)} \;:\; M(x,w,y) = 1\right\}\right| < K(x,w)/2 \end{aligned}$$

NSBP *is defined as the class of sets in* prNSBP*.*

Analogously to Circuit-Count-Gap, it is possible to define a promise problem such that prNSBP is the class of all promise problems reducible to that problem.

Theorem 11. prNSBP = prSBP *and thus* NSBP = SBP*.*

Proof. The basic idea is to modify the computation so the number of accepted y's (for a given w) is multiplied by an efficiently computable factor, so as to shift the threshold to be close to some value that does not depend on w (indeed, does not even depend on x). Then as before we can use amplification so that the gap swamps out the effect of the nondeterministic w. However, there is a slight wrinkle to iron out: If $K(x,w) = 0$ then we cannot shift the threshold by multiplying it by something. But in this case x is automatically a YES instance, so if we happen to observe $K(x,w) = 0$ then we can just accept.

We proceed with the formal proof. We reduce an arbitrary $\Pi \in$ prNSBP (with associated M, K, and p) to Circuit-Count-Gap. Given $x \in \{0,1\}^n$, we let p denote $p(n)$ and we assume without loss of generality that $0 \leq K(x,w) \leq 2^p$ for all $w \in \{0,1\}^p$. We construct a circuit $C : \{0,1\}^p \times \left(\{0,1\}^p \times \{0,1\}^{p+3}\right)^{2p} \to \{0,1\}$ by

$$C\big(w, (y_i, z_i)_{i=1}^{2p}\big) \;=\; \begin{cases} 1 & \text{if } \; K(x,w) = 0 \;\vee\; \forall i \left[M(x,w,y_i) = 1 \;\wedge\; z_i < \lceil 2^{p+3}/K(x,w) \rceil\right] \\ 0 & \text{otherwise} \end{cases}$$

where each z_i is viewed as the binary representation of a nonnegative integer, and we let $k = (2^{p+3})^{2p}$. We show the following two things.

$$\begin{aligned} x \in \Pi_{\text{YES}} &\implies (C,k) \in \textsc{Circuit-Count-Gap}_{\text{YES}} \\ x \in \Pi_{\text{NO}} &\implies (C,k) \in \textsc{Circuit-Count-Gap}_{\text{NO}} \end{aligned}$$

For the YES case, consider the w whose existence is guaranteed by Definition 10. If $K(x,w) = 0$ then C accepts $\big(w, (y_i, z_i)_{i=1}^{2p}\big)$ for every choice of $(y_i, z_i)_{i=1}^{2p}$, and thus the total number of accepted inputs is at least $(2^{2p+3})^{2p} \geq k$. On the other hand, assume $K(x,w) > 0$. Since $M(x,w,y_i) = 1$ holds for at least $K(x,w)$ choices of y_i, and $z_i < \lceil 2^{p+3}/K(x,w) \rceil$ holds for $\lceil 2^{p+3}/K(x,w) \rceil$ choices of z_i, the conjunction of these holds for at least $K(x,w) \cdot \lceil 2^{p+3}/K(x,w) \rceil \geq 2^{p+3}$ choices of (y_i, z_i). Hence C accepts $\big(w, (y_i, z_i)_{i=1}^{2p}\big)$ for at least $(2^{p+3})^{2p} = k$ choices of $(y_i, z_i)_{i=1}^{2p}$, and thus the total number of accepted inputs is also at least k. For the NO case, consider an arbitrary w. We must have $K(x,w) > 0$ (since if $K(x,w) = 0$ then automatically $x \in \Pi_{\text{YES}}$ by Definition 10). Since $M(x,w,y_i) = 1$ holds for less than $K(x,w)/2$ choices of y_i, and $z_i < \lceil 2^{p+3}/K(x,w) \rceil$ holds for $\lceil 2^{p+3}/K(x,w) \rceil$ choices of z_i, the conjunction of these holds for less than $(K(x,w)/2) \cdot \lceil 2^{p+3}/K(x,w) \rceil \leq \big(2^{p+3} + K(x,w)\big)/2 \leq 2^{p+3} \cdot (5/8)$ choices of (y_i, z_i) (using $K(x,w) \leq 2^p$). Hence C accepts $\big(w, (y_i, z_i)_{i=1}^{2p}\big)$ for less than $\big(2^{p+3} \cdot (5/8)\big)^{2p}$ choices of $(y_i, z_i)_{i=1}^{2p}$. Summing over w, the total number of accepted inputs is less than $2^p \cdot \big(2^{p+3} \cdot (5/8)\big)^{2p} = k \cdot (25/32)^p < k/2$. □

5 Discussion

The chief open problem is to determine the complexity of estimating the min-entropy of distributions that are generated by 2-local samplers. Is it SBP-complete?

It is clear that for any NP relation with a parsimonious reduction from 3-Sat, approximate counting is as hard as possible (i.e., the associated promise problem is SBP-complete). The problem Monotone-2-Sat has no such parsimonious reduction (since every instance is satisfiable), but we have nevertheless shown that it is as hard as possible to approximately count. Are there other natural NP relations for which approximate counting is as hard as possible for nontrivial reasons?

Also, we would like to point out some further implications our proof of Theorem 4 has on the complexity of Monotone-2-Sat. It follows from our argument that the exact threshold analogue of Monotone-2-Sat-Count-Gap (distinguishing at least k satisfying assignments from less than k satisfying assignments) is PP-complete (under mapping reductions), and that #Monotone-2-Sat (where the goal is to output the number of satisfying assignments) is #P-complete (under one-query reductions). The former did not seem to be known. The latter is well-known, but the only proof we could find mentioned in the literature is due to Valiant [Val79a, Val79b] and is based on the #P-completeness of computing the permanent. Our argument is much more elementary and demonstrates that this heavy machinery is not needed for the #P-completeness of #Monotone-2-Sat.

Acknowledgments

I thank Oded Goldreich and anonymous reviewers for their comments.

References

[AAB+10] Scott Aaronson, Barış Aydınlıoğlu, Harry Buhrman, John Hitchcock, and Dieter van Melkebeek. A note on exponential circuit lower bounds from derandomizing Arthur-Merlin games. Technical Report TR10-174, Electronic Colloquium on Computational Complexity, 2010.

[AGHK11] Barış Aydınlıoğlu, Dan Gutfreund, John Hitchcock, and Akinori Kawachi. Derandomizing Arthur-Merlin games and approximate counting implies exponential-size lower bounds. *Computational Complexity*, 20(2):329–366, 2011.

[AK01] Vikraman Arvind and Johannes Köbler. On pseudorandomness and resource-bounded measure. *Theoretical Computer Science*, 255(1-2):205–221, 2001.

[App14] Benny Applebaum. *Cryptography in Constant Parallel Time*. Information Security and Cryptography. Springer, 2014.

[BBM11] Nayantara Bhatnagar, Andrej Bogdanov, and Elchanan Mossel. The computational complexity of estimating MCMC convergence time. In *Proceedings of the 15th International Workshop on Randomization and Computation*, pages 424–435, 2011.

[BGM06] Elmar Böhler, Christian Glaßer, and Daniel Meister. Error-bounded probabilistic computations between MA and AM. *Journal of Computer and System Sciences*, 72(6):1043–1076, 2006.

[BMV08] Andrej Bogdanov, Elchanan Mossel, and Salil Vadhan. The complexity of distinguishing Markov random fields. In *Proceedings of the 12th International Workshop on Randomization and Computation*, pages 331–342, 2008.

[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.

[DGGJ03] Martin Dyer, Leslie Ann Goldberg, Catherine Greenhill, and Mark Jerrum. The relative complexity of approximate counting problems. *Algorithmica*, 38(3):471–500, 2003.

[DGRV11] Zeev Dvir, Dan Gutfreund, Guy Rothblum, and Salil Vadhan. On approximating the entropy of polynomial mappings. In *Proceedings of the 2nd Innovations in Computer Science Conference*, pages 460–475, 2011.

[DRV12] Yevgeniy Dodis, Thomas Ristenpart, and Salil Vadhan. Randomness condensers for efficiently samplable, seed-dependent sources. In *Proceedings of the 9th Theory of Cryptography Conference*, pages 618–635, 2012.

[DW12] Anindya De and Thomas Watson. Extractors and lower bounds for locally samplable sources. *ACM Transactions on Computation Theory*, 4(1), 2012.

[FGL$^+$96] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996.

[FIKU08] Lance Fortnow, Russell Impagliazzo, Valentine Kabanets, and Christopher Umans. On the complexity of succinct zero-sum games. *Computational Complexity*, 17(3):353–376, 2008.

[Gol06] Oded Goldreich. On promise problems: A survey. In *Essays in Memory of Shimon Even*, pages 254–290. Springer, 2006.

[GS86] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *Proceedings of the 18th ACM Symposium on Theory of Computing*, pages 59–68, 1986.

[GSV99] Oded Goldreich, Amit Sahai, and Salil Vadhan. Can statistical zero knowledge be made non-interactive? or On the relationship of SZK and NISZK. In *Proceedings of the 19th International Cryptology Conference*, pages 467–484, 1999.

[Hås99] Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[JVV86] Mark Jerrum, Leslie Valiant, and Vijay Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

[KRC00] Valentine Kabanets, Charles Rackoff, and Stephen Cook. Efficiently approximable real-valued functions. Technical Report TR00-034, Electronic Colloquium on Computational Complexity, 2000.

[KRVZ11] Jesse Kamp, Anup Rao, Salil Vadhan, and David Zuckerman. Deterministic extractors for small-space sources. *Journal of Computer and System Sciences*, 77(1):191–220, 2011.

[Kup09] Greg Kuperberg. How hard is it to approximate the Jones polynomial? *CoRR*, abs/0908.0512, 2009.

[KvM02] Adam 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.

[LP02] Rune Lyngsø and Christian Pedersen. The consensus string problem and the complexity of comparing hidden Markov models. *Journal of Computer and System Sciences*, 65(3):545–569, 2002.

[MU02] Elchanan Mossel and Christopher Umans. On the complexity of approximating the VC dimension. *Journal of Computer and System Sciences*, 65(4):660–671, 2002.

[MV05] Peter Bro Miltersen and N. V. Vinodchandran. Derandomizing Arthur-Merlin games using hitting sets. *Computational Complexity*, 14(3):256–279, 2005.

[San89] Miklos Santha. Relativized Arthur-Merlin versus Merlin-Arthur games. *Information and Computation*, 80(1):44–49, 1989.

[Sha11] Ronen Shaltiel. An introduction to randomness extractors. In *Proceedings of the 38th International Colloquium on Automata, Languages and Programming*, pages 21–41, 2011.

[Sin93] Alistair Sinclair. *Algorithms for Random Generation and Counting: A Markov Chain Approach*. Progress in Theoretical Computer Science. Birkhäuser, 1993.

[Sto85] Larry Stockmeyer. On approximation algorithms for #P. *SIAM Journal on Computing*, 14(4):849–861, 1985.

[SU06] Ronen Shaltiel and Christopher Umans. Pseudorandomness for approximate counting and sampling. *Computational Complexity*, 15(4):298–341, 2006.

[TV00] Luca Trevisan and Salil Vadhan. Extracting randomness from samplable distributions. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, pages 32–42, 2000.

[TVZ05] Luca Trevisan, Salil Vadhan, and David Zuckerman. Compression of samplable sources. *Computational Complexity*, 14(3):186–227, 2005.

[Vad12] Salil Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1-3), 2012.

[Vad13] Salil Vadhan. *A Study of Statistical Zero-Knowledge Proofs*. Springer, 2013.

[Val79a] Leslie Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

[Val79b] Leslie Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[Ver92] Nikolai Vereshchagin. On the power of PP. In *Proceedings of the 7th Structure in Complexity Theory Conference*, pages 138–143, 1992.

[Vio11] Emanuele Viola. Extractors for circuit sources. In *Proceedings of the 52nd IEEE Symposium on Foundations of Computer Science*, pages 220–229, 2011.

[Vio12] Emanuele Viola. Extractors for Turing-machine sources. In *Proceedings of the 16th International Workshop on Randomization and Computation*, pages 663–671, 2012.