ELSEVIER

Theoretical Computer Science 191 (1998) 215–218

Theoretical Computer Science

Note

Computational indistinguishability: algorithms vs. circuits

Oded Goldreich[a], Bernd Meyer[b,*]

[a] *Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel*

[b] *Pegasusstraße 14, D-85716 Unterschleißheim, Germany*

Received July 1996
Communicated by O. Watanabe

Abstract

We present a simple proof of the existence of a probability ensemble with tiny support which cannot be distinguished from the uniform ensemble by any recursive computation. Since the support is tiny (i.e., sub-polynomial), this ensemble can be distinguished from the uniform ensemble by a (non-uniform) family of small circuits. It also provides an example of an ensemble which cannot be (recursively) distinguished from the uniform by one sample, but can be so distinguished by two samples. In case we only wish to fool probabilistic polynomial-time algorithms the ensemble can be constructed in super-exponential time.

1. Introduction

Computational indistinguishability, introduced by Goldwasser and Micali [4] and defined in full generality by Yao [7], is a central concept of complexity theory. Two probability ensembles, $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$, where both X_n and Y_n range over $\{0,1\}^n$, are said to be indistinguishable by a complexity class if for every machine M in the class the difference $\mathrm{Prob}(M(X_n)=1) - \mathrm{Prob}(M(Y_n)=1)$ is a negligible function in n (i.e., decreases faster than $1/p(n)$ for any positive polynomial p).

It has been known for a while (cf. [7, 5, 3]) that there exist probability ensembles which are statistically far from the uniform ensemble and yet computationally indistinguishable from it: In [7, 5] indistinguishability is with respect to (probabilistic) polynomial-time algorithms, whereas in [3] indistinguishability is with respect to polynomial-size circuits. A simple proof is via the probabilistic method: Fix any function $d:\{0,1\}^n \mapsto \{0,1\}$, and select at random $\mathrm{O}(t/\varepsilon^2)$ strings of length n. Then, by Hoefding's inequality, with probability greater than $1-2^{-t}$ the average value of d

* Corresponding author. E-mail: Bernd.Meyer@munich.netsurf.de.

0304-3975/98/$19.00
PII S0304-3975(97)00162-X

over this sample will be within $\pm\varepsilon$ of the average over the entire domain $\{0,1\}^n$. Substituting for d the characteristic function of an arbitrary Turing machine M, and using a standard enumeration of Turing machines, it is possible to diagonalize against all Turing machines. Thus, for any super-polynomial function $s:\mathbb{N}\mapsto\mathbb{N}$ there exists a probability ensemble, with support[1] size bounded by $s(\cdot)$, which is indistinguishable from the uniform ensemble by any Turing machine. Clearly, time bounds on the distinguishing machines yield obvious bounds on the time required to construct the ensemble. Furthermore, the same argument can be applied to probabilistic machines as well as to non-uniform families of circuits (e.g., all polynomial-size circuits).

In [6], two probability ensembles, having sparse but disjoint supports, are shown to be indistinguishable by probabilistic polynomial-time algorithms. Specifically, the support size is n^2 and the distinguishing probability is exponentially vanishing in n. It seems that the argument in [6] cannot yield either a support of size $o(n\log n)$ nor zero distinguishing probability. Here we present a simpler proof of the following stronger result.

Proposition 1 (Main result). *Let $\mathcal{M}$ be an enumeration of halting* (probabilistic) *Turing machines, and $t:\mathbb{N}\mapsto\mathbb{N}$ be any non-decreasing and unbounded function. Then, there exists a probability ensemble, $\{R_n\}$, so that, for every $n\in\mathbb{N}$:*

(i) *The support of R_n has size at most $t(n)+1$.*

(ii) *For each one of the first $t(n)$ machines in $\mathcal{M}$, denoted M,*

$$\mathrm{Prob}(M(R_n)=1)=\mathrm{Prob}(M(U_n)=1),$$

where U_n denotes the uniform distribution over $\{0,1\}^n$.

Furthermore, in case $\mathcal{M}$ is the set of probabilistic polynomial-time machines, the distribution R_n can be constructed in time $e(n)$, where $e:\mathbb{N}\mapsto\mathbb{N}$ is any function which grows faster than $2^{\mathrm{poly}(n)}$.

Observe that the ensemble $\{R_n\}$ is $\mathcal{M}$-indistinguishable from $\{U_n\}$ in a strong sense: For each Turing machine $M\in\mathcal{M}$ the difference $\mathrm{Prob}(M(R_n)=1)-\mathrm{Prob}(M(U_n)=1)$ is non-zero only on finitely many n's. As immediate corollaries we get:

Corollary 2. *There exists a probability ensemble, $\{R_n\}$, which is indistinguishable from the uniform ensemble by probabilistic polynomial-time machines but is distinguishable from it by a family of polynomial-size circuits.*

Proof. Let $t(n)$ be bounded by a polynomial in n (e.g., $t(n)=n$ or $t(n)=\log_2\log_2 n$ will do), and let $\{R_n\}$ be as guaranteed by Proposition 1. Thus, the ensemble $\{R_n\}$ is indistinguishable from the uniform by probabilistic polynomial-time machines. On the other hand, the following (non-uniform) family of polynomial-size circuits distinguishes

[1] The support of a probability distribution is the set of strings which are assigned non-zero probability under this distribution.

the ensemble from the uniform ensemble: The nth circuit incorporates the support of R_n and outputs 1 if and only if the input is in the support. □

Corollary 3. *There exists a probability ensemble, $\{R_n\}$, which is indistinguishable from the uniform ensemble by probabilistic polynomial-time machines but is distinguishable from it by a polynomial-time algorithm which gets two* (independently drawn) *samples from the distribution.*

Proof. Again, using a polynomially bounded $t(n)$, define the ensemble $\{R_n\}$ as in Proposition 1. Thus, $\{R_n\}$ is indistinguishable from the uniform ensemble by probabilistic polynomial-time machines. On the other hand, the following polynomial-time algorithm which obtains two samples distinguishes the ensemble from the uniform ensemble: The algorithm, which gets two samples, outputs 1 if and only if both samples are identical. □

We comment that both [1, 6] present a result related to the last corollary. Specifically, they present two ensembles, each with at most two n-bit strings in their support, for which all single-sample algorithms have vanishing distinguishing probability, whereas a simple two-sample algorithm has constant distinguishing probability. Note that in the corollary above the size of the support of R_n is small (e.g., $\log\log n$) but not a constant. Yet, the distinguishing probability based on a single sample is zero (for all but a finite set of indices of the ensembles).

We stress that all results in the paper are absolute (i.e., do not require any unproven assumptions). On the other hand, the fact that the ensembles are not constructible in polynomial-time is unavoidable, since analogous results for polynomial-time constructible (samplable) ensembles imply the existence of one-way functions (cf. [2]).

2. Proof of main result

Suppose that you want to construct a distribution with small support which fools (i.e., looks random to) a single machine, denoted M. Then all you need is two strings, $x, y \in \{0,1\}^n$, so that

$$\mathrm{Prob}(M(x)=1) \leqslant \mathrm{Prob}(M(U_n)=1), \tag{1}$$

$$\mathrm{Prob}(M(y)=1) \geqslant \mathrm{Prob}(M(U_n)=1). \tag{2}$$

Fixing these x and y, there exists an $\alpha \in [0,1]$ so that defining the distribution R_n so that $R_n = x$ with probability α and $R_n = y$ otherwise, you get

$$\mathrm{Prob}(M(R_n)=1) = \mathrm{Prob}(M(U_n)=1).$$

Thus, machine M cannot distinguish R_n from U_n.

All that is needed for proving the main result is to generalize the argument so that we can fool t machines *simultaneously*. To this end, consider the 2^n (t-dimensional) vectors corresponding to the probabilities that each of the t machines outputs 1 on each of the strings in $\{0,1\}^n$. Specifically, the vector associated with $x \in \{0,1\}^n$ has $\mathrm{Prob}(M_i(x)=1)$ in its ith component, where M_i is the ith machine (that we are trying to fool). Observe that the average of these vectors, denoted $\bar{v}$, is a vector of probabilities with $\mathrm{Prob}(M_i(U_n)=1)$ as its ith component. The average vector $\bar{v}$ is in the convex hull of all 2^n former vectors. Since the dimension of the span of all 2^n vectors is bounded by t there exist $t+1$ vectors which (also) have $\bar{v}$ in their convex hull. Let $v_1,\dots,v_{t+1}$ denote a set of such $t+1$ vectors. Then, by definition, there exists $\alpha_1,\dots,\alpha_{t+1}$ non-negative and summing up to 1, so that the vector $\sum_{j=1}^{t+1} \alpha_j v_j$ equals the vector $\bar{v}$. Using the x_j's corresponding to these vectors with the coefficients α_j's, we get the desired distribution. Specifically, we define R_n so that $\mathrm{Prob}(R_n = x_j) = \alpha_j$ for $j = 1,\dots,t+1$. Clearly, for $i = 1,\dots,t$,

$$\mathrm{Prob}(M_i(R_n)=1) = \sum_{j=1}^{t+1} \alpha_i \cdot \mathrm{Prob}(M_i(x_j)=1) = \mathrm{Prob}(M_i(U_n)=1).$$

References

[1] M.J. Fischer, S.A. Paleologou, On the indistinguishability of probabilistic ensembles, unpublished manuscript, 1994.

[2] O. Goldreich, A note on computational indistinguishability, Inform. Process. Lett. 34 (1990) 277–281.

[3] O. Goldreich, H. Krawczyk, On sparse pseudorandom ensembles, Random Structures and Algorithms, 3 (2) (1992) 163–174.

[4] S. Goldwasser, S. Micali, Probabilistic encryption, J. Comput. System Sci. 28 (2) (1984) 270–299 (Preliminary version in 14th ACM Symp. on Theory of Computing, 1982).

[5] L.A. Levin, mid 1980s, private communication.

[6] B. Meyer, Constructive separation of classes of indistinguishable ensembles, 9th IEEE Symp. on Structure in Complexity Theory (1994) 198–204.

[7] A.C. Yao, Theory and application of trapdoor functions, 23rd IEEE Symp. on Foundations of Computer Science, 1982, pp. 80–91.