Constructing Small Sample Spaces Satisfying Given Constraints

Daphne Koller[*]
e-mail: daphne@cs.stanford.edu

Nimrod Megiddo[†]
e-mail: megiddo@almaden.ibm.com

Abstract

Abstract. The subject of this paper is finding small sample spaces for joint distributions of n discrete random variables. Such distributions are often only required to obey a certain limited set of constraints of the form $Pr(E) = \pi$. We show that the problem of deciding whether there exists any distribution satisfying a given set of constraints is NP-hard. However, if the constraints are consistent, then there exists a distribution satisfying them which is supported by a "small" sample space (one whose cardinality is equal to the number of constraints). For the important case of *independence constraints*, where the constraints have a certain form and are consistent with a joint distribution of n independent random variables, a small sample space can be constructed in polynomial time. This last result is also useful for de-randomizing algorithms. We demonstrate this technique by an application to the problem of finding large independent sets in sparse hypergraphs.

1. Introduction

The probabilistic method of proving existence of combinatorial objects has been very successful (see, for example, [13, 15]). The underlying idea is as follows. Consider a finite set Ω whose elements are classified as "good" and "bad". Suppose we wish to prove existence of at least one "good" element within Ω. The proof proceeds by constructing a probability distribution f over Ω and showing that the probability of picking a good element is positive. Probabilistic proofs often yield randomized algorithms for constructing a good element. In particular, many randomized algorithms are a special case of this technique, where the "good" elements are those sequences of random bits leading to a correct answer.

It is often desirable to replace the probabilistic construction by a deterministic one, or to *de-randomize* an algorithm. Obviously, this can be done by completely enumerating the sample space Ω until a good element is found. Unfortunately, the sample space is typically exponential in terms of the size of the problem; for example, the obvious sample space of n independent coin tosses contains 2^n points.

More precisely, let $X_1, \ldots, X_n$ be discrete random variables with a finite range. For simplicity, we assume that $X_1, \ldots, X_n$ all have the same range $\{0, \ldots, r-1\}$ (although not necessarily the same distribution). Our constructions can easily be extended to variables with different ranges. The *probability space* associated with these variables is $\Omega = \{0, \ldots, r-1\}^n$. A *distribution* is a map $f : \Omega \to [0, 1]$ such that $\sum_{\boldsymbol{x} \in \Omega} f(\boldsymbol{x}) = 1$.

[*] Department of Computer Science, Stanford University, Stanford, CA 94305; and IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120.

[†] IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120; and School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel.

Research supported in part by ONR Contract N00014-91-C-0026 and by the Air Force Office of Scientific Research (AFSC), under Contract F49620-91-C-0080.

We define the set $S(f) = \{\boldsymbol{x} \in \Omega \mid f(\boldsymbol{x}) > 0\}$ to be the *essential sample space of* f.

Given a distribution f involved in a probabilistic proof, only the points in $S(f)$ need to be considered in our search for a good point in Ω. Moreover, if it easy to recognize whether a point $\boldsymbol{x}$ in $S(f)$ is good for a particular input, then it suffices to search any subset of $S(f)$ which is guaranteed to contain a good point for each possible input. Adleman [2] shows that for any distribution f supporting an algorithm in RP, there exists a space $S \subseteq S(f)$ of polynomial size that contains a good point for every possible input. The proof of this fact is not constructive, and therefore cannot be used for de-randomizing algorithms.

A common technique for constructing a smaller space to search is to construct a different distribution with a "small" (polynomial) essential sample space that can be searched exhaustively, as outlined above. The new distribution must agree with the original one sufficiently so that the correctness proof of the algorithm remains valid. The correctness proof often relies on certain assumptions about the distribution; that is, the distribution is assumed to satisfy certain constraints. A *constraint* is an equality of the form

$$\Pr(Q) = \sum_{\boldsymbol{x} \in Q} f(\boldsymbol{x}) = \pi \ ,$$

where $Q \subseteq \Omega$ is an *event* and $0 \leq \pi \leq 1$. If the randomness requirements of an algorithm are completely describeable as a set of constraints, and the new distribution satisfies all of them, then the algorithm remains valid under the new distribution. Moreover, no new analysis is needed. In other cases, the new distribution may only approximately satisfy the constraints, and it is necessary to check that the analysis still holds.

In almost all cases, the original distribution is constructed based on *independent* random variables $X_1, \ldots, X_n$. Thus, all the constraints are satisfied by such a distribution. In many cases, however, full independence is not necessary. In particular, quite often the constraints are satisfied by d-wise independent distributions for some small d. Most of the previous work has focused on constructing approximations to such distributions.

Joffe [9] first demonstrated a construction of a joint distribution of n d-wise independent random variables with an essential sample space of cardinality $O(n^d)$. Luby [10] and Alon, Babai, and Itai [1] generalize Joffe's construction to non-uniform distributions. In many cases, these constructions only approximately satisfy the required constraints; that is, the distributions are d-wise independent, but the probabilities $\Pr(X_i = b)$ may differ from the corresponding probabilities in the original distribution. This construction results in a sample space of polynomial size for any fixed d. It is shown in [7] that the cardinality of a sample space of a joint distribution of n d-wise independent random bits[1] is $\Omega(n^{[d/2]})$. Thus, these constructions are close to optimal in this case. Moreover, sample spaces of polynomial size exist for d-wise independent distributions only if d is fixed.

Naor and Naor [12] showed how to circumvent this lower bound by observing that *ϵ-independent* (or *nearly independent*) distributions often suffice. In other words, it suffices that a constraint, stating that a particular event must occur as if the variables were independent, be satisfied to within ϵ. We point out that this is also a form of approximation, as defined above. Naor and Naor demonstrate a construction of sample spaces for ϵ-independent distributions over random bits, whose size is polynomial in n and in $1/\epsilon$. These constructions are polynomial for $\epsilon = 1/\text{poly}(n)$; for such values of ϵ, the ϵ-independence constraints are meaningful for subsets of size up to[2] $O(\log n)$. Therefore, we obtain a polynomial-size sample space which is nearly d-wise independent for $d = O(\log n)$ (as com-

[1] We use the term *random bits* to denote binary valued uniformly distributed random variables.

[2] Consider a distribution over random bits, and some subset of k of the variables. The "correct" probability of any event prescribing values to all the variables in this subset is $1/2^k$. For $k = O(\log n)$, this value is $O(\epsilon)$. So for larger k, all such constraints are essentially subsumed by constraints corresponding to smaller subsets of the variables.

pared to the lower bound of $\Omega(n^{\log n})$ for truly d-wise independent sample spaces). Simplified constructions with similar properties were provided by Alon *et. al* [3]. Azar, Motwani and Naor [5] later generalized these techniques to uniform distributions over non-binary random variables. Finally, Even *et. al* [8] presented constructions for arbitrary nearly d-wise independent distributions over non-binary random variables.

A different type of technique was introduced by Berger and Rompel [6] and by Motwani, Naor and Naor [11]. This technique can be used to derandomize certain RNC algorithms where d, the degree of independence required, is polylogarithmic in n. The technique works, however, only for certain types of problems, and does not seem to generalize to larger degrees of independence.

Schulman [14] took a different approach towards the construction of sample spaces which require $O(\log n)$-wise independence. He observed that in many cases, only certain *d-neighborhoods* (sets of d variables) must be independent. Schulman constructs sample spaces satisfying this property whose size is 2^d times the greatest number of neighborhoods to which any variable belongs. In particular, for polynomially many neighborhoods each of size $O(\log n)$, this construction results in a polynomial-size sample space. His construction works only for random bits, and for a maximum neighborhood size $O(\log n)$.

In order to improve on these results, we view the problem from a somewhat different perspective. Instead of placing upper bounds on the degree of independence required by the algorithm, we examine the set of precise constraints that are required in order for the algorithm to work. We then construct a distribution satisfying these constraints exactly. In many cases, this approach yields a much smaller sample space, as we explain below.

We begin by showing a connection between the number of constraints and the size of the resulting sample space. We show in Section 2 that for any set $\mathcal{C}$ of such constraints, if $\mathcal{C}$ is *consistent*, *i.e.*, $\mathcal{C}$ is satisfied by some distribution f, then there exists a distribution f' also satisfying $\mathcal{C}$ such that $|S(f')| \leq |\mathcal{C}|$. That is, there exists a distribution for which the cardinality of the essential sample space is not more than the number of constraints. As before, if the constraints represent all the assumptions about f made by a proof, the proof will also hold for f'. If $S(f')$ is sufficiently small, we can exhaustively enumerate it, resulting in a deterministic construction. The proof of the existence theorem includes a technique for constructing f'; however, the technique requires exponential time and is thus not useful. We justify the exponential behavior of this algorithm by showing that even for a set $\mathcal{C}$ of very simple constraints, the problem of recognizing whether there exists a distribution f satisfying $\mathcal{C}$ is NP-complete.

Our goal is to define a type of constraints for which a small sample space can be constructed directly from the constraints in polynomial time. As we observed, the distributions that are most often used in probabilistic proofs are ones where $X_1, \ldots, X_n$ are independent random variables. Such a distribution is determined entirely by the probabilities $\{p_{ib} = \Pr(X_i = b) : i = 1, \ldots, n;\ b = 0, \ldots, r-1\}$. In the course of such a probabilistic proof, the distribution is assumed to satisfy various constraints, which often has a form as follows. An *independence constraint* is one which forces the probability of a certain assignment of values to some subset of the variables to be as if the variables are independent. That is, for a fixed set of p_{ib}'s, $\{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$, and $b_1, \ldots, b_k \in \{0, \ldots, r-1\}$, the constraint

$$\Pr(\{X_{i_1} = b_1\ , \ldots,\ X_{i_k} = b_k\}) = \prod_{j=1}^{k} p_{i_j b_j}$$

is the *independence constraint* corresponding to the event $Q = \{X_{i_1} = b_1\ , \ldots,\ X_{i_k} = b_k\}$, and is denoted by $I(Q)$. Obviously, if $X_1, \ldots, X_n$ are independent random variables then their joint distribution satisfies all the independence constraints. Note that d-wise independence can easily be represented in terms of constraints of this type: the variables $X_1, \ldots, X_n$ are d-wise independent if and only if all the independence constraints $I(\{X_{i_1} = b_1, \ldots, X_{i_d} = b_d\})$ are satisfied, where $i_1, \ldots, i_d \in \{1, \ldots, n\}$ are distinct indices and $b_1, \ldots, b_d \in \{0, \ldots, r-1\}$. In other words, $X_1, \ldots, X_n$ are d-wise independent if and only if every event defined over a neighborhood of size d has the same probability as if the variables were independent.

Let $\mathcal{C}$ be a set of independence constraints defined using a fixed set of p_{ib}'s as above. In Section 3 we present the main result of this paper, which shows how to construct in polynomial time a distribution satisfying $\mathcal{C}$ with an essential sample space of cardinality $|\mathcal{C}|$. We note that the distribution produced by our technique is typically not uniform. Therefore, we cannot in general use our construction to reduce the number of uniformly distributed random bits required to generate the desired distribution.

Our construction has a number of advantages. First, the distributions generated always satisfy the constraints precisely. This fact allows one to use precisely the same correctness proof for the new distribution as for the old one, without the need for a new analysis. Moreover, the size of the sample space in all the nearly independent constructions [3, 5, 8, 12] depends polynomially on $1/\epsilon$ (where ϵ is the approximation factor). Our precise construction does not have this term. Previously, precise distributions were unavailable for many interesting distributions. In particular, our approach can construct sample spaces of cardinality $O((rn)^d)$ for any set of n r-valued, d-wise independent random variables. For fixed d, this construction requires polynomial time. It has been argued [8] that probability distributions over non-uniform non-binary random variables are important. To our knowledge, this is the first technique which allows the construction of exact distributions of d-wise independent variables with arbitrary p_{ib}'s.

The main advantage of our construction is that the size of the sample space depends only on the number of constraints actually used. Except for Schulman's approach [14], all other sample spaces are limited by requiring that all neighborhoods of a particular size be independent (or nearly independent). As Schulman points out, in many cases only certain neighborhoods are ever relevant, thus enabling a further reduction in the size of the sample space. However, Schulman's approach still requires the sample space to satisfy all the independence constraints associated with the relevant neighborhoods. This restricts his construction to neighborhoods of maximal size[3] $O(\log n)$. With our construction we can deal with neighborhoods of any size, as long as the number of relevant constraints is limited.

For example, an algorithm may randomly choose edges in a graph by associating a binary random variable with each edge. An event whose probability may be of interest is "no edge adjacent to a node v is chosen". Using the other approaches (even Schulman's), the neighborhood size would be the maximum degree Δ of a node in the graph; the relevant sample space would then grow as 2^{Δ}. Using our approach, there is only one event per node, resulting in a sample space of size n (the number of nodes in the graph).

In this example, the constraints depend on the edge structure of the input graph. In general, our construction depends on the specific constraints associated with a particular instance of the input. Therefore, unlike most sample space constructions, our construction cannot be prepared in advance. This property, combined with the fact that our algorithm is sequential, means that it cannot be used to convert RNC algorithms into NC ones.

In Section 4 we show an example of how our technique can be applied to de-randomization of algorithms. We discuss the problem of finding a large independent set in a d-uniform hypergraph. The underlying randomized algorithm, described in [1], was de-randomized in the same paper for fixed values of d. It was later de-randomized also for $d = O(\text{polylog} n)$ in [6] and [11]. We show how this algorithm can be de-randomized for any d. We point out that a sequential deterministic polynomial time solution for the independent set problem in hypergraphs exists [4]. However, the de-randomization of this algorithm using our technique serves to demonstrate its unique power.

[3] Moreover, as we have observed, Schulman's construction works only for random bits.

2. Existence of small essential sample spaces

Let $\mathcal{C} = \{[\Pr(Q_k) = \pi_k]\}_{k=1,\ldots,c}$ be a set of constraints such that $[\Pr(\Omega) = 1] \in \mathcal{C}$.

Definition 2.1. A set $\mathcal{C}$ of constraints is *consistent* if there exists some distribution f satisfying all the members of $\mathcal{C}$.

Definition 2.2. A distribution f that satisfies $\mathcal{C}$ is said to be *manageable* if $|S(f)| \leq c = |\mathcal{C}|$.

Theorem 2.3. *If $\mathcal{C}$ is consistent, then $\mathcal{C}$ is satisfied by a manageable distribution.*

Proof: Let $\mathcal{C}$ be as above, and recall that $c = |\mathcal{C}|$. We describe a distribution f satisfying $\mathcal{C}$ as a non-negative solution to a set of linear equations. Let $\boldsymbol{\pi} \in \mathbb{R}^c$ denote the vector $(\pi_k)_{k=1,\ldots,c}$. Recall that $\Omega = \{0, \ldots, r-1\}^n$; let $m = |\Omega| = r^n$, and let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ denote the points of Ω. The variable v_ℓ will represent the probability $f(\boldsymbol{x}_\ell)$. Let $\boldsymbol{v}$ be the vector $(v_\ell)_{\ell=1,\ldots,m}$. A constraint $\Pr(Q_k) = \pi_k$ can be represented as the linear equation

$$\sum_{\ell=1}^{m} a_{k\ell} v_\ell = \pi_k \ ,$$

where

$$a_{k\ell} = \begin{cases} 1 & \text{if } \boldsymbol{x}_\ell \in Q_k \\ 0 & \text{otherwise .} \end{cases}$$

Thus, the constraints in $\mathcal{C}$ can be represented by a system $\boldsymbol{A}\boldsymbol{v} = \boldsymbol{\pi}$ of linear equations (where $\boldsymbol{A}$ is the matrix $(a_{k\ell})_{k\ell}$. Since $\mathcal{C}$ is assumed to be consistent, there is a distribution f satisfying $\mathcal{C}$. Therefore, for $v_\ell = f(\boldsymbol{x}_\ell)$, the vector $\boldsymbol{v}$ is a nonnegative solution to this system. A classical theorem in linear programming asserts that under these conditions, there exists a basic solution to this system. That is, there exists a vector $\boldsymbol{v}' \geq \boldsymbol{0}$ such that $\boldsymbol{A}\boldsymbol{v}' = \boldsymbol{\pi}$ and the columns of $\boldsymbol{A}$, $\boldsymbol{A}_{*j}$, such that $v'_j > 0\}$ are linearly independent. Let f' be the distribution corresponding to this solution vector $\boldsymbol{v}'$. Since the number of rows in the matrix is c, the number of linearly independent columns is also at most c. Therefore, the number of positive indices in $\boldsymbol{v}'$, which is precisely $|S(f')|$, is at most $c = |\mathcal{C}|$. ∎

This theorem can be proven constructively based on the following standard algorithm, which begins with a distribution vector $\boldsymbol{v}$, and removes points from the sample space one at a time, resulting in a manageable distribution vector $\boldsymbol{v}'$. Througout the algorithm, let $S(\boldsymbol{v})$ denote the set of indices $\{j \,:\, v_j > 0\}$. Intuitively, these indices represent points in the essential sample space of the distribution represented by $\boldsymbol{v}$.

Algorithm 2.4.

```
While $\{\boldsymbol{A}_{*j} : j \in S(\boldsymbol{v})\}$ are linearly dependent:

1. Find a nonzero vector $\boldsymbol{u} \in R^m$ such that
       $u_j = 0$ for every $j \notin S(\boldsymbol{v})$, and
       $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{0}$.

2. Find some $t \in \mathbb{R}$ such that
       $\boldsymbol{v} + t\boldsymbol{u} \geq \boldsymbol{0}$, and
       $v_j + tu_j = 0$ for some $j \in S(\boldsymbol{v})$.

3. Replace $\boldsymbol{v} \leftarrow \boldsymbol{v} + t\boldsymbol{u}$.
```

Alternatively, the manageable distribution can be computed directly from the constraints using a linear programming algorithm which computes basic solutions. Unfortunately, since Algorithm 2.4 handles the variables one at a time, it runs in time which is polynomial in $m = r^n$. Similarly, a linear programming algorithm runs in time polynomial in m. Thus, both these techniques are exponential in n.

The exponential behavior of these algorithms can be justified by considering the problem of deciding whether a given set of constraints $\mathcal{C}$ is consistent; that is, does there exist a distribution f satisfying the constraints in $\mathcal{C}$? For arbitrary constraints, the representation of the events can be very long, causing the input size to be unreasonably large. We therefore restrict attention to *simple constraints.*

Definition 2.5. We say that a constraint $\Pr(Q) = \pi$ is *k-simple* if there exist $i_1, \ldots, i_k \in \{1, \ldots, n\}$ and $b_1, \ldots, b_k \in \{0, \ldots, r-1\}$ such that $Q = \{[X_{i_1} = b_1], \ldots, [X_{i_k} = b_k]\}$. A constraint is *simple* if it is k-simple for some k.

Note that the natural representation of the event as a simple constraint requires space which is at most linear in n, whereas the number of points in the event is often exponential in n (for example, a 1-simple constraint contains r^{n-1} points). We assume throughout that simple constraints are represented compactly (in linear space).

It turns out that the consistency problem is NP-hard, even when restricted to 2-simple constraints:

Proposition 2.6. *The problem of recognizing whether a set $\mathcal{C}$ of 2-simple constraints is consistent is* NP*-hard.*

Proof: The proof is based on a simple reduction from the 3-colorability problem. See the full paper for details. ∎

In order to prove a matching upper bound, we again need to make a simple assumption about the representation of the input.

Definition 2.7. An event Q is said to be *polynomially checkable* if membership of any point $\boldsymbol{x} \in \Omega$ in Q can be checked in time polynomial in n.

Proposition 2.8. *If all the constraints in $\mathcal{C}$ pertain to polynomially checkable events, then the consistency of $\mathcal{C}$ can be decided in nondeterministic polynomial time (in terms of $|\mathcal{C}|$ and n).*

Proof: The algorithm guesses a subset $T \subset \Omega$ of cardinality $|\mathcal{C}|$. It then constructs in polynomial time a system of equations corresponding to the constraints in $\mathcal{C}$ restricted to the variables in T (the other variables are set to 0). Given the initial guess, this system can be constructed in polynomial time, since for each constraint and each point in T it takes polynomial time to check whether the point appears in the constraint. The algorithm then attempts to find a nonnegative solution to this system. Such a solution exists if and only if there exists a manageable distribution whose essential sample space is (contained in) T. By Theorem 2.3, we know that a set of constraints is consistent if and only if it is satisfied by a manageable distribution; that is, a distribution over some sample space T of cardinality not greater than $|\mathcal{C}|$. Therefore, $\mathcal{C}$ is consistent if and only if one of these subsystems has a nonnegative solution. ∎

Since simple constraints are always polynomially checkable (using the appropriate representation), we obtain the following theorem.

Theorem 2.9. *For an arbitrary set $\mathcal{C}$ of simple constraints, the problem of recognizing the consistency of $\mathcal{C}$ is* NP*-complete.*

3. Independence constraints

An important special case was already discussed in the introduction. Suppose all the members of $\mathcal{C}$ are independence constraints arising from a known fixed set of values

$$\{\Pr(X_i = b) = p_{ib} : i = 1, \ldots, n; b = 0, \ldots, r-1\},$$

such that $\sum_{b=0}^{r-1} p_{ib} = 1$ for all i, and $p_{ib} \geq 0$ for all i, b. In this case we can construct a distribution satisfying $\mathcal{C}$ over a sample space of cardinality $c = |\mathcal{C}|$; the construction is done in time polynomial in c, r and n.

We first define the concept of a *projected event.* Consider an event

$$Q = \{[X_{i_1} = b_1], \ldots, [X_{i_k} = b_k]\}.$$

We assume without loss of generality that $i_1 < i_2 < \ldots < i_k$; that is, the variables in the constraints are listed in order of increasing index (we make this assumption for every event mentioned in this section). Let ℓ ($1 \leq \ell \leq n$) be an integer and denote by $q = q(\ell)$ the maximal index such that $i_q \leq \ell$. The *ℓ-projection* of Q is defined as

$$\Pi_\ell(Q) = \{X_{i_1} = b_1 \;, \dots, \; X_{i_k} = b_q\} \; .$$

Intuitively, the ℓ-projection of a constraint is its restriction to the variables $X_1, \dots, X_\ell$. For example, if Q is $\{X_1 = 0, X_4 = 1, X_7 = 1\}$, then $\Pi_3(Q) = \{X_1 = 0\}$ and $\Pi_4(Q) = \{X_1 = 0, X_4 = 1\}$. Analogously, we call $I(\Pi_\ell(Q))$ the ℓ-projection of the constraint $I(Q)$. Finally, for a set of constraints $\mathcal{C}$, $\Pi_\ell(\mathcal{C})$ is the set of the ℓ-projections of the constraints in $\mathcal{C}$.

We now define recursively a sequence of distributions $f_1, \dots, f_n$, such that for each ℓ ($\ell = 1, \dots, n$), the following conditions hold:

(i) f_ℓ is a distribution on $\{0, \dots, r-1\}^\ell$,

(ii) f_ℓ satisfies $\Pi_\ell(\mathcal{C})$,

(iii) $|S(f_\ell)| \leq c$.

The distribution f_n is clearly the desired one.

We begin by defining for all $b \in \{0, \dots, r-1\}$

$$f_1((b)) = p_{1b} \; .$$

This clearly satisfies all the requirements.

Now, assume that $f_{\ell-1}$ (for $\ell > 1$) satisfies the above requirements, and define an intermediate distribution g_ℓ by:

$$g_\ell(x_1, \dots, x_{\ell-1}, b) = f_{\ell-1}(x_1, \dots, x_{\ell-1}) \cdot p_{\ell b} \quad (1)$$

for $b = 0, \dots, r-1$.

Lemma 3.1. *If $f_{\ell-1}$ satisfies $\Pi_{\ell-1}(\mathcal{C})$, then g_ℓ satisfies $\Pi_\ell(\mathcal{C})$.*

Proof: Suppose $I(Q)$ is an arbitrary constraint in $\mathcal{C}$, where $Q = \{X_{i_1} = b_1 \;, \dots, \; X_{i_k} = b_k\}$. For simplicity, denote $Q_j = \Pi_j(Q)$ ($j = 1, \dots, n$). Let r be the maximal index such that $i_r \leq \ell - 1$. By the assumption,

$$f_{\ell-1}(Q_{\ell-1}) = \prod_{j=1}^{r} p_{i_j b_j} \; .$$

We distinguish two cases:
Case I: Q mentions the variable X_ℓ. In this case, $i_{r+1} = \ell$, and

$$\begin{aligned} Q_\ell &= \{X_{i_1} = b_1, \dots, X_{i_r} = b_r, X_{i_{r+1}} = b_{r+1}\} \\ &= \{(x_1, \dots, x_{\ell-1}, b_{r+1}) : (x_1, \dots, x_{\ell-1}) \in Q_{\ell_1}\} \; . \end{aligned}$$

Therefore:

$$\begin{aligned} g_\ell(Q_\ell) &= \sum_{(x_1,\dots,x_{\ell-1}) \in Q_{\ell-1}} g_\ell(x_1, \dots, x_{\ell-1}, b_{r+1}) \\ &= \sum_{(x_1,\dots,x_{\ell-1}) \in Q_{\ell-1}} f_{\ell-1}(x_1, \dots, x_{\ell-1}) \cdot p_{\ell b_{r+1}} \\ &= f_{\ell-1}(Q_{\ell-1}) \cdot p_{\ell b_{r+1}} \\ &= \prod_{j=1}^{r} p_{i_j b_j} \cdot p_{\ell b_{r+1}} == \prod_{j=1}^{r+1} p_{i_j b_j} \; . \end{aligned}$$

Thus, g_ℓ satisfies the constraint $I(Q_\ell)$.
Case II: Q does not mention the variable X_ℓ. In this case,

$$\begin{aligned} Q_\ell &= \{X_{i_1} = b_1, \dots, X_{i_r} = b_r\} \\ &= \{(x_1, \dots, x_\ell) : (x_1, \dots, x_{\ell-1}) \in Q_{\ell-1}, \\ &\qquad x_\ell \in \{0, \dots, r-1\}\} \; . \end{aligned}$$

Therefore:

$$\begin{aligned} g_\ell(Q_\ell) &= \sum_{b \in \{0,\dots,r-1\}} \sum_{(x_1,\dots,x_{\ell-1}) \in Q_{\ell-1}} g_\ell(x_1, \dots, x_{\ell-1}, b) \\ &= \sum_{b \in \{0,\dots,r-1\}} \sum_{(x_1,\dots,x_{\ell-1}) \in Q_{\ell-1}} f_{\ell-1}(x_1, \dots, x_{\ell-1}) p_{\ell b} \\ &= \sum_{b \in \{0,\dots,r-1\}} p_{\ell b} f_{\ell-1}(Q_{\ell-1}) \\ &= f_{\ell-1}(Q_{\ell-1}) = \prod_{j=1}^{r} p_{i_j b_j} \; . \end{aligned}$$

Again, g_ℓ satisfies the constraint $I(Q_\ell)$. ▮

If $|S(f_{\ell-1})| \leq c$, then $|S(g_\ell)| \leq rc$, since each point with positive probability in $S(f_{\ell-1})$ yields at most r points with positive probabilities in $S(g_\ell)$. Thus, g_ℓ satisfies requirements (i) and (ii), but may not satisfy requirement (iii). But g_ℓ is a nonnegative solution to the system of linear equations defined by the ℓ-projections of the constraints in $\mathcal{C}$. Therefore, we may use Algorithm 2.4 to reduce the cardinality of the essential sample space to c. Let f_ℓ be the resulting distribution. It clearly satisfies all three requirements. We thus obtain the following theorem:

Theorem 3.2. *Given a set of independence constraints, we can construct a manageable distribution f satisfying $\mathcal{C}$ in time $O(rnc^2)$.*

Proof: The distribution f_n constructed as above is clearly a manageable distribution satisfying $\mathcal{C}$. The construction takes $n-1$ iterations. Each iteration requires at most $O(rc)$ operations to create g_ℓ from $f_{\ell-1}$, and at most $O(rc^2)$ arithmetic operations for running Algorithm 2.4 to reduce g_ℓ to f_ℓ. Therefore, the entire algorithm runs in $O(rnc^2)$ operations. Note that the number of operations does not depend on the magnitudes of the numbers in the input. ∎

Our algorithm can easily be extended to operate on random variables with ranges of different sizes. Let r_i be the number of values in the range of X_i. The sample space of g_ℓ will consist of vectors $(x_1,\ldots,x_{\ell-1},b)$ where $(x_1,\ldots,x_{\ell-1}) \in S(f_{\ell-1})$ and $b \in \{0,\ldots,r_\ell\}$. Then

$$|S(g_\ell)| \le r_\ell|\mathcal{C}| \ .$$

The proof goes through as before, but the number of operations in iteration i is $O(r_ic^2)$. The total number of operations is $O((\sum_{i=1}^n r_i)c^2) = O(rnc^3)$, where $r = \max\{r_1,\ldots,r_n\}$. The cardinality of the resulting sample space is still $|\mathcal{C}|$.

The assumption that the p_{ib}'s are known is important in view of the following theorem, which states that if this is not the case, it is NP-hard to verify whether all of a given set of constraints are independence constraints. This result holds even for binary valued random variables ($r=2$).

Theorem 3.3. *It is* NP*-hard to recognize whether for a given set of simple constraints $\mathcal{C}$ there exists a set $P = \{p_{i0}, p_{i1} \in [0,1] : i = 1,\ldots,n; p_{i0}+p_{i1} = 1\}$ such that all the members of $\mathcal{C}$ are independence constraints relative to P.*

Proof: We prove the theorem by reduction from SAT. A CNF formula φ is satisfiable iff its negation $\neg\varphi$ is not valid. Moreover, $\neg\varphi$ is a boolean formula in disjunctive normal form (DNF), whose length is linear in the length of φ (using de Morgan's laws). We can therefore use a reduction to the non-validity of a DNF formula. Given a DNF formula φ in the variables $u_1,\ldots,u_n$, we build a set $\mathcal{C}$ as follows. Let $\psi = y_1 \wedge \cdots \wedge y_k$ be a disjunct in φ, where $y_j \in \{u_{i_j}, \bar{u}_{i_j}\}$ $(j = 1,\ldots,k)$, and $i_1 < \cdots < i_k$. We associate with ψ a constraint

$$\Pr(Q_\psi) = 0 \ ,$$

where

$$\begin{aligned} Q_\psi &= \{[X_{i_1} = b_1],\ldots,[X_{i_k} = b_k]\} \\ b_j &= \begin{cases} 1 & \text{if } y_j = u_{i_j} \\ 0 & \text{if } y_j = \bar{u}_{i_j} \ . \end{cases} \end{aligned}$$

Consider an assignment θ of truth values to the variables $u_1,\ldots,u_n$, and a set P as in the statement of the theorem. We say that θ and P *correspond* if for all i, $p_{i0} = 0$ iff $\theta(u_i) =$ *false*. It is easy to find for any truth assignment θ a corresponding set P, and vice versa. Clearly, if θ and ψ correspond, the constraint associated with Q_ψ is an independence constraint with respect to P if and only if $\theta(\psi) =$ *false*. Similarly, all the constraints in $\mathcal{C}$ are independence constraints with respect to P if and only if $\theta(\varphi) =$ *false*. Therefore, φ is not valid iff $\mathcal{C}$ is a set of independence constraints for some set P. ∎

It is not clear that the problem of Theorem 3.3 is in NP. The set P relative to which a given $\mathcal{C}$ is a set of independence constraints might contain irrational numbers even if all the input numbers are rational.

Example 3.4. Consider the problem of constructing a distribution over the binary-valued variables X_1, X_2, and X_3 satisfying

$$\begin{aligned} \Pr(\{X_1 = 1, X_2 = 1\}) &= \tfrac{1}{2} \\ \Pr(\{X_1 = 1, X_3 = 1\}) &= \tfrac{1}{2} \\ \Pr(\{X_2 = 1, X_3 = 1\}) &= \tfrac{1}{2} \ . \end{aligned}$$

These are independence constraints only with respect to $p_{11} = p_{21} = p_{31} = \frac{1}{\sqrt{2}}$.

Nevertheless, in most practical cases, the p_{ib}'s are part of the specification of the algorithm. Thus, it is usually reasonalbe to assume that they are known.

4. De-randomizing algorithms

In this section we demonstrate how the technique of Section 3 can be used to de-randomize algorithms. We present three progressively improving ways in which the technique can be applied. For the sake of simplicity and for ease of comparison, we will base our analysis on a single problem. This is the problem of finding large independent sets in sparse hypergraphs. The problem description and the randomized algorithm for its solution are taken from Alon, Babai, and Itai [1]. We point out that a deterministic polynomial-time algorithm for this problem is known [4].

A *d-uniform hypergraph* is a pair $\mathcal{H} = (V, \mathcal{E})$ where $V = \{v_1, \ldots, v_n\}$ is a set of *vertices* and $\mathcal{E} = \{E_1, \ldots, E_m\}$ is a collection of subsets of V, each of cardinality d, which are called *edges*. A subset $U \subseteq V$ is said to be *independent* if it contains no edge. For simplicity, we restrict attention to d-uniform hypergraphs; a similar analysis goes through in the general case.

Consider the following randomized algorithm (k will be defined later).

Algorithm 4.1.

```
1. Construct a random subset $R$ of $V$.
   For each vertex $v_i \in V$:
       put $v_i$ in $R$ with probability $p = 3k/n$.
2. Modify $R$ into an independent set $U$.
   For each edge $E_j \in \mathcal{E}$ such that $E_j \subseteq R$:
       remove from $R$ one arbitrary vertex $v_i \in E_j$.
```

Proposition 4.2 (Alon, Babai, Itai) : *If $\mathcal{H} = (V, \mathcal{E})$ is a d-uniform hypergraph with n vertices and m edges, then for* $k = (1/18)(n^d/m)^{1/(d-1)}$ *Algorithm 4.1 finds an independent set of cardinality exceeding k with probability greater than* $\frac{1}{2} - \frac{3}{k}$.

Proof: For each i, let X_i be the random variable that equals 1 if $v_i \in R$ and 0 otherwise. For each edge $E_j \in \mathcal{E}$, let Y_j the random variable that equals 1 if $E_j \subseteq R$ and 0 otherwise. The cardinality of R is $|R| = \sum_{i=1}^{n} X_i = X$, so $E(X) = np = 3k$.

- If the X_i's are pairwise independent, then the variance of X is

$$\sigma^2(X) = \sum_{i=1}^{n} \sigma^2(X_i) = np(1-p) < np = 3k \ . \tag{2}$$

  Thus, using Chebychev's inequality,

$$\Pr(X \le 2k) \le \frac{\sigma^2(X)}{k^2} < \frac{3}{k} \ .$$

- If the X_i's are d-wise independent then for every $j = 1, \ldots, m$,

$$E(Y_j) = \Pr\left(\bigcap_{i \in E_j} \{X_i = 1\}\right) = p^d \tag{3}$$

  Let $Y = \sum_{j=1}^{m} Y_j$ denote the number of edges contained in R. Computation shows that $\Pr(Y \ge k) < \frac{1}{2}$.

If R contained at least $2k$ vertices after the first stage in the algorithm, and at most k vertices were removed in the second stage, then the independent set constructed by the algorithm has cardinality of at least k. This has probability

$$\ge \Pr\left(\{Y < k\} \cap \{X \ge 2k\}\right) > \frac{1}{2} - \frac{3}{k} \ .$$

∎

De-randomization I

The de-randomization procedure of Alon, Babai, and Itai [1] is based on constructing a joint distribution of d-wise independent variables which approximates the joint d-wise independent distribution of variables X_i for which $\Pr(X_i) = 3k/n$ ($i = 1, \ldots, n$). It is then necessary to analyze this approximate distribution. Our technique provides exactly the required distribution, so that no further analysis is needed. As we explained in the introduction, this can be done by considering the set $\mathcal{C}^I$ of the constraints:[4]

$$\{I(\{X_{i_1} = b_1, \ldots, X_{i_d} = b_d\}) \; : \\ i_1, \ldots, i_d \in \{1, \ldots, n\},\; b_1, \ldots, b_d \in \{0,1\}\} \; .$$

The number of these constraints is $|\mathcal{C}^I| = \binom{n}{d}2^d = O((2n)^d)$. For fixed d, this number is polynomial in n, resulting in a sample space of polynomial size. Therefore, the algorithm runs in polynomial time, including both the phase of constructing the sample space and the phase of running Algorithm 4.1 on each point of this space until a sufficiently large independent set is found.

De-randomization II

A closer examination of the proof reveals that not all the $\binom{n}{d}$ neighborhoods of cardinality d have to be independent. In order for equation (3) to hold, it suffices that only the X_i's associated with vertices in the same edge be independent. If $E_j = \{v_{i_1}, \ldots, v_{i_d}\}$, let $\mathcal{C}_j$ denote the set of 2^d independence constraints

$$\{I(\{X_{i_1} = b_1, \ldots, X_{i_d} = b_d\}) \; : b_1, \ldots, b_d \in \{0,1\}\} \; .$$

On the other hand, in order for equation (2) to hold, the choices must still be pairwise independent. Let $\mathcal{C}^2$ denote the set of $4\binom{n}{2}$ constraints

$$\{I(\{X_{i_1} = b_1, X_{i_2} = b_2\}) \; : \\ i_1, i_2 \in \{1, \ldots, n\},\; b_1, b_2 \in \{0,1\}\} \; .$$

Thus, the following set of constraints suffices:

$$\mathcal{C}^{II} = \mathcal{C}^2 \cup \bigcup_{E_j \in \mathcal{E}} \mathcal{C}_j \; .$$

More precisely, if the set $\mathcal{C}^{II}$ is satisfied then the proof of Proposition 4.2 goes through, and the resulting sample space must contain a point which is good for this hypergraph. Since the number of constraints is

$$|\mathcal{C}^{II}| = |\mathcal{C}^2| + \sum_{E_j \in \mathcal{E}} |\mathcal{C}_j| = 4\binom{n}{2} + m2^d \; ,$$

this results in a polynomial-time algorithm for $d = O(\log n)$, which applies to a larger class of graphs than the one presented in [1]. At first glance, it seems that as we have polynomially many neighborhoods of logarithmic size, Schulman's technique [14] can also be used to extend the results in [1]. However, his approach is limited to uniform distributions, so it does not apply to this algorithm. The results of Berger and Rompel [6] and of Motwani, Naor, and Naor [11], however, provide a polynomial-time algorithm for $d = O(\text{polylog}n)$. Their results use a completely different technique, and cannot be extended to handle larger values of d.

De-randomization III

A yet closer examination of the proof of Proposition 4.2 reveals that Equation (3) does not require complete independence of the neighborhood associated with the edge E_j. It suffices to constrain the probability of the event "all the vertices in E_j are in R" (the event corresponding to the random variable Y_j in the proof). That is, for $E_j = \{v_{i_1}, \ldots, v_{i_d}\}$, we need only the independence constraint over the event:

$$Q_j = \{X_{i_1} = 1, \ldots, X_{i_d} = 1\} \; .$$

This is a simple event which defines an independence constraint of the type to which our technique applies. We conclude that the following set of constraints suffices for the analysis of Proposition 4.2 goes through:

$$\mathcal{C}^{III} = \mathcal{C}^2 \cup \{I(Q_j) : E_j \in \mathcal{E}\} \; .$$

The number of constraints

$$|\mathcal{C}^{III}| = 4\binom{n}{2} + m$$

[4] Recall that the indices $i_1, \ldots, i_d$ are assumed to be distinct and sorted in increasing order.

is polynomial in the size of the problem $(n+m)$ regardless of d. Therefore, this results in a deterministic polynomial-time algorithm for finding large independent sets for arbitrary uniform hypergraphs.

5. Conclusions and open questions

We have presented a new approach to constructing distributions with small sample spaces. Our technique constructs a distribution tailored explicitly to the required constraints. The construction is based on an explicit representation of the constraints as a set of linear equations over the distribution. It enables us to construct sample spaces for arbitrary distributions over discrete random variables, which are precise (not approximations) and sometimes considerably smaller than sample spaces constructed using previously known techniques.

A number of open questions arise immediately from our results.

- Schulman's approach constructs a sample space whose size depends not on the total number of neighborhoods involved in constraints, but on the maximum number of such neighborhoods in which a particular variable appears. Can the size of the sample space in our approach be similarly reduced to depend on the maximum number of constraints in which a variable participates.
- We mentioned in the introduction that the nature of our approach generally prevents a precomputation of the manageable distribution. However, our approach shows the existence of manageable distributions which are useful in general contexts. For exfample, for every n, d, and p, we show the existence of a d-wise independent distribution over n binary random variables such that $\Pr(X_i = 1) = p$ for all i. It would be useful to come up with an explicit construction for this case.
- Our technique constructs distributions that precisely satisfy a given set of arbitrary independence constraints. It is natural to ask if our results can be improved by only requiring the distribution to approximately satisfy these constraints. In particular, maybe we can construct approximate distributions faster, or in parallel, or over smaller sample spaces. We note that the construction of [1, 9, 10] can be viewed as finding a distribution that precisely satisfies the d-wise independence constraints but approximately satisfies the constraints of the form $\Pr(X_i = b)$. In contrast, the nearly-independent constructions [3, 5, 8, 12] can be viewed as approximately satisfying the different d-wise independence constraints. Thus, they can all be viewed as providing an answer to this question for certain types of constraint-sets $\mathcal{C}$ and certain restrictions on which constraints can be approximated.
- Combined with our inability to precompute the distribution, the sequential nature of our construction prevents its use for derandomization of parallel algorithms. Parallelizing the construction could open up many application areas for this approach.

Acknowledgements

The authors wish to thank Yossi Azar, David Karger, and Moni Naor for useful comments.

References

[1] N. Alon, L. Babai, and A. Itai, “A fast and simple randomized parallel algorithm for the maximal independent set problem,” *Journal of Algorithms*, **7** (1986) 567–583.

[2] L. Adleman, “Two theorems on random polynomial time,” in: *Proceedings of the 19th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Los Angeles, 1978, pp. 75–83.

[3] N. Alon, O. Goldreich, J. Hastad, and R. Peralta, “Simple constructions of almost k-wise

independent random variables," in: *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Los Angeles, 1990, pp. 544–553.

[4] N. Alon, Private communication.

[5] Y. Azar, R. Motwani, and J. Naor. "Approximating arbitrary probability distributions using small sample spaces," unpublished manuscript.

[6] B. Berger and J. Rompel. "Simulating $(\log^c n)$-wise independence in NC," in: *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Los Angeles, 1989, pp. 2–7.

[7] B. Chor, O. Goldreich, J. Hastad, J. Friedman, S. Rudich, and R. Smolensky, "t-resilient functions," in: *Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Los Angeles, 1985, pp. 396–407, 1985.

[8] G. Even, O. Goldreich, M. Luby, N. Nisan, and B. Veličkovi'c, "Approximations of general independent distributions," in: *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1992, pp. 10–16.

[9] A. Joffe, "On a set of almost deterministic k-independent random variables," *Annals of Probability* **2** (1974) 161–162.

[10] M. Luby, "A simple parallel algorithm for the maximal independent set problem," *SIAM Journal on Computing* **15** (1986) 1036–1053.

[11] R. Motwani, J. Naor, and M. Naor, "The probabilistic method yields deterministic parallel algorithms," in: *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Los Angeles, 1989, pp. 8–13.

[12] J. Naor and M. Naor, "Small-bias probability spaces: Efficient constructions and applications, in: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, ACM, New York, 1990, pp. 213–223.

[13] P. Raghavan, "Probabilistic construction of deterministic algorithms: Approximating packing integer problems," *J. Comp. Sys. Sci.*, **37** (1988) 130–143.

[14] L. J. Schulman, "Sample spaces uniform on neighborhoods," in: *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, ACM, (1992) pp. 17–25.

[15] J. Spencer, *Ten Lectures on the Probabilistic Method*, Society for Industrial and Applied Mathematics, 1987.