Randomized Methods in Computation*
Lecture 3: Small Pairwise-Independent Sample Spaces

Notes taken by Lev Faivishevsky and Sergey Khristo

Spring 2001

We ...solemnly Publish and Declare that...
these are ...free and independent...
(Declaration of Independence)
Action of Second Continental Congress, July 4, 1776

Summary: Often in applications that use randomness (for example coin tosses) we do not need all these tosses to be independent, instead it may suffice that they are k-independent for some k. In the case of independent variables the probability space is at least of the size $|S|^m$, where m is the number of variables in the sequence and S is the sample space for an individual variable. In this lecture we present a construction for pairwise independent sample space of the size $\max(|S|, m)^2$ and its extension to k-wise independent case. This will allow us to *derandomize* the algorithm that finds assignment for 3CNF that satisfies at least 7/8 clauses, presented on the first lecture, just by means of exhaustive search.

1 Construction of k-wise independent sample space

Suppose we have sample space of the form $S^m = S \times S \times \ldots \times S$, where S is a finite set, with uniform distribution on it. Without loss of generality we can think of S as a set of integers $\{1, \ldots, |S|\}$.

Any element of S^m is a sequence of m elements from S, denoted $(s_1, \ldots, s_m)$. Clearly, every two components, taken alone, also have uniform distribution. Note, that if the pair $(x_1, x_2) \in X \times X$ is uniformly distributed on $X \times X$ then x_1 and x_2 are independent and uniformly distributed on X.

We are seeking for the subspace Ω that is substantially smaller than S^m, but the uniform distribution on Ω induces uniform distribution on any two coordinates. In the construction for Ω, presented below, we achieve $|\Omega| \simeq \max(|S|, m)^2$. This allow us, in some sense, to *save* randomness.

The construction that is described in detail below is a nice example but there are some technicalities that should be taken into account when working with it. They are discussed at the end of this lecture.

Suppose, for the beginning, that $m \leq |S|$. This restriction will be waived later. We will try to associate some algebraic structure to S; for example, by treating S as a finite field $\mathcal{F}$. Finite fields exist only of sizes which are prime powers. The case when the size of S is not a prime power will be discussed later.

* Lecture Notes for a course given by Oded Goldreich at the Weizmann Institute of Science, Israel.

Let us now define such a sample space Ω:

$$\begin{aligned}\Omega &\stackrel{\text{def}}{=} \{\text{The set of all linear polynomials over } \mathcal{F}\} = \{ax+b|(a,b)\in\mathcal{F}^2\}\\ &\equiv \{(a\cdot 1+b, a\cdot 2+b,\ldots,a\cdot m+b)|(a,b)\in\mathcal{F}^2\}.\end{aligned}$$

The first line above gives a succinct representation of a set of m-long sequences over $\mathcal{F}$, whereas the second line gives an explicit representation of the same sequences.

Note that given a field $\mathcal{F}$, the space Ω can be easily sampled: you just pick two random elements from the field and get an element of Ω.

Let us denote by $\Pr_{x\in_R X}$ the probability over the set X with uniform measure on it. Write $p_{a,b}(i) = a\cdot i+b$ (the linear polynomial in i with coefficients a and b).

Claim 1.1 *For every $i\neq j$ from $[m]\subseteq\mathcal{F}$ and for every α and β from $\mathcal{F}$*

$$\Pr_{(a,b)\in_R\mathcal{F}^2}[p_{a,b}(i)=\alpha\wedge p_{a,b}(j)=\beta] = \frac{1}{|\mathcal{F}|^2}.$$

In other words, uniform measure on Ω results in pairwise independent uniform sample space.

Proof: Let us write formally

$$\Pr_{(a,b)\in_R\mathcal{F}^2}[p_{a,b}(i)=\alpha\wedge p_{a,b}(j)=\beta] = \frac{|\{(a,b)|\ p_{a,b}(i)=\alpha\wedge p_{a,b}(j)=\beta\}|}{|\mathcal{F}|^2}.$$

We have to show that the nominator equals one. Indeed, it is equal to the number of pairs (a,b) such that the following system of two equations is satisfied.

$$\begin{aligned} ai+b&=\alpha\\ aj+b&=\beta\end{aligned}$$

Thinking of a and b as variables and of i, j, α and β as constants, we conclude that if i is not equal j then this system has unique solution because the matrix

$$\begin{pmatrix} i & 1\\ j & 1\end{pmatrix}$$

has full rank. ∎

Observe that the whole construction is independent of the field we use and one can use any field that he finds convenient. For example, if the size of S is prime then modular arithmetic is a good choice.

Note that the shown construction is not 3-wise independent. First, we do not have uniform (and independent) distribution on any *three* coordinates, since the there can be no solution for the system of the equations as in the proof of claim.

More generally, it is not possible to have a k-wise independent sample space that is smaller then $|S|^k$, because any fixed k coordinates are uniformly distributed and every value of $(x_1,\ldots,x_k)$ has some nonzero probability hence it is impossible to have less then $|S|^k$ elements in this set.

After a little meditating it becomes clear how to make our construction produce k-wise independent variables. For this it is enough to consider in the above definition of Ω polynomials of degree less than k. Specifically, we define

$$\begin{aligned}\Omega' &\stackrel{\text{def}}{=} \{\text{The set of polynomials of degree less then } k \text{ over } \mathcal{F}\}\\ &= \{\overline{c} = (c_0, \ldots, c_{k-1}) : p_{\overline{c}} = \sum_{i=0}^{k-1} c_i x^i\}\\ &\equiv \{(p_{\overline{c}}(1), \ldots, p_{\overline{c}}(m)) : \overline{c} \in \mathcal{F}^k\}.\end{aligned}$$

Again, the first lines give a succinct representation of a set of m-long sequences over $\mathcal{F}$, whereas the last line gives an explicit representation of the same sequences.

Recall that sample space is called k-wise independent if on any fixed k locations we have uniform distribution. Analogously to Claim 1.1, we have the following:

Claim 1.2 *For every different $i_1, \ldots, i_k$ from $[m] \subset \mathcal{F}$ and for every $\alpha_1, \ldots, \alpha_m$ from $\mathcal{F}$*

$$\Pr_{\overline{c} \in_R \mathcal{F}^k}[\forall j = 1 \ldots k, p_{\overline{c}}(i_j) = \alpha_{i_j}] = \left(\frac{1}{|\mathcal{F}|}\right)^k.$$

Proof: Similar to what we had in the previous claim,

$$\Pr_{\overline{c} \in_R \mathcal{F}^k}[\forall j = 1 \ldots k, p_{\overline{c}}(i_j) = \alpha_j] = \frac{|\{\overline{c}|\forall j = 1 \ldots k, p_{\overline{c}}(i_j) = \alpha_j\}|}{|\mathcal{F}|^k}$$

The nominator is equal the number of solutions to the following system:

$$\begin{cases} c_0 + c_1 i_1 + \ldots + c_{k-1} i_1^{k-1} = \alpha_1 \\ \vdots \\ c_0 + c_1 i_k + \ldots + c_{k-1} i_k^{k-1} = \alpha_k \end{cases}$$

Again one should think of c_i's as variables in the linear system.

To see that in fact there is exactly one solution let us write the system in the matrix form:

$$\begin{pmatrix} 1 & i_1 & i_1^2 & \ldots & i_1^{k-1} \\ 1 & i_2 & i_2^2 & \ldots & i_2^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & i_k & i_1^k & \ldots & i_k^{k-1} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{k-1} \end{pmatrix} = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{k-1} \end{pmatrix}$$

The matrix in the above system is a Van der Monde matrix and it is full rank (since $i_l \neq i_{l'}$ for every $l \neq l'$). ■

This construction is optimal in the size of Ω, but remember we are under the conditions that $m \leq |S|$ and that the size of S is must prime power (it allows us to treat S as a field and make all that claims about uniqueness of the solution of a linear system). The next section addresses these issues and shows how to deal with fields from the computational point of view.

2 Annoying technicalities

In this section we will cope with several technical pitfalls, that can potentially spoil our construction. These potential difficulties are:

1. The size of S may be not a prime power. It means that direct representation S as a finite field is impossible in this case.

2. What should we do if $m > |S|$?

3. How do we deal with fields, i.e. how can we define the sample subspace and how can we operate in it?

First of all, let's address the first problem. A general recipe is to consider substantially larger set S' with a "nice" embedding of S into S'. The meaning of nice is that each element of S appears in S' about the same number of times. So if we make a construction for S', it will remain valid for S also because there will be some good mapping from S' to S. More formally, we should have some mapping $\mu : S' \to S$ with the following properties:

$$\forall a, b \in S \quad |\mu^{-1}(a)| \approx |\mu^{-1}(b)|$$

The formula says that each image should have nearly the same number of preimages (it means that the mapping μ has to be "regular"). Actually, it is better that each element of S will have the same number of representatives in S'. If this condition doesn't hold then (exact) uniform distribution in S' wouldn't induce uniform distribution in S. Vice versa, if each element in S has exactly the same number of representatives in S' then uniform distribution in S' will induce uniform distribution in S via this mapping μ. It should be noticed that small discrepancy in this approximate equality like:

$$\forall a, b \in S \quad |\mu^{-1}(a)| = |\mu^{-1}(b)| \pm 1$$

doesn't disturb the validity of our constructions by much (especially if $|S'|/|S|$ is large; indeed, the bigger S' the better the approximation but the cost comes in terms of complexity which grows with S'). Hence whenever the initial space S doesn't satisfy us, we may switch to a larger S'. The same trick will help us further, in coping with the second problem.

Now we will give a similar solution to the second problem. In order to be more clear, we'll do it through an example. Suppose, we need to build sample space of 3-wise independent elements over S={0,1}. More specifically, we want to construct the sample space of 3-wise independent sequences of length m with elements of S. We'll show that size of this sample has to be m^3 only. By Claim 1.2 (with $k = 3$) we would be able to do it if $|S|$ were bigger than m. Thus, we ought to take

$$S' = \{0,1\}^l, \quad \text{where} \quad l = \lceil \log_2 m \rceil$$

Note, that this condition guarantees that $|S'| \geq m$, because

$$|S'| = 2^l = 2^{\lceil \log_2 m \rceil} \geq m$$

Finally one should mention, that $|S'|$ is a prime power. The size of the whole construction will be

$$|S'|^3 \approx (2l)^3 \approx m^3$$

The price that we have paid for the trick is that our constructed sample subspace has rather big size $\approx m^3$. The last step that should be done is to define the mapping $\mu : S' \to S$ that in our case implies

$$\mu : \{0,1\}^l \to \{0,1\}$$

The simple mapping may be just taking the first bit of an element in $\{0,1\}^l$ as the element's image in $\{0,1\}$. All needed properties of μ are satisfied. Generally, defining such a mapping is a relatively easy task, because we impose only the following restrictions on μ: it should be easy to compute and the mapping should be regular (remember, this means that each image should have nearly the same number of preimages).

The last technical problem to be treated is the handling of finite fields. We have assigned algebraic structure to constructed sample subspace. But how can we handle it? This issue clearly consists of several significant parts:

1. How do we build the sample space?
2. How do we explicitly write the elements of it?
3. How do we perform operations with the elements (like addition and multiplication)?

The answer to these questions amounts to providing an algorithm for constructing sample spaces. We will claim that this algorithm exists "modulo the field". In case of a field of prime cardinality, the meaning of the sentence "modulo the field" is that we are given the prime p. The general case will be considered later.

Claim 2.1 *Given prime p and number $m \geq p$ we may construct in* poly(m)*-time the sample space* Ω*:*

$$\Omega = \{\text{The set of all linear polynomials over field } \mathcal{F}\} \equiv \{ax+b | (a,b) \in \mathcal{F}^2\}$$

$$\equiv \{(a\cdot 1+b, a\cdot 2+b, \ldots, a\cdot m+b) | (a,b) \in \mathcal{F}^2\}.$$

Proof: In order to prove the claim we will explicitly give the algorithm that outputs p^2 such sequence of length m each. The algorithm will run in time polynomial in $p*m$.

The algorithm consists of two nested loops.

```
for a,b ∈ F
    for i = 1 to m
        output (a · i + b) mod p).
```

We can see that running time of the algorithm is $O(p^2)$. Also if we want to generate a specific element and we know which one and a specific coordinate in it, we are also able to do it very efficiently. ■

Oded's Note: *What I tried to explain is that in general, we may be given an arbitrary set S and an integer m and need to construct a pairwise (or k-wise) independent space over S. In case $|S|$ is a prime and $|S| \geq m$, the above claim provides the answer, but in general $|S|$ may not be a prime (or even a prime power) or $m > |S|$. In such a case, we shall resort to the technique presented above; that is, construct a pairwise (or k-wise) independent space over some sufficiently large S', and map S' to S. But in such a case, we have to find S' (which has to be of prime (or prime power) cardinality) by ourselves. We may choose to find S' of prime cardinality or find S' of cardinality p^e where p is a small prime and e is an integer. In the first case, we face the problem of finding a large prime, and if the desired prime is huge then we are in real trouble (becuase we don't know of an efficient deterministic algorithm; see details below). In such a case, we turn to the second alternative, and use S' of cardinality p^e, when p is samll (e.g., $p = 2$). Such a prime poer is easy to find, but operating with a field of that cardinality is more involved.*

About the complexity of finding primes. Certainly, given an integer N, we can find a prime in the interval $[N, 2N]$ by using the trivial algorithm that scans all candiadtes and tries to factor each of them. This algorithm is deterministic and run in time polynomial in N, which is fine when N is not too big. A randomized algorithm can find such a prime in time polynomial in the length of the binary representation of N (i.e., $\text{poly}\log N$), but randomized algorithms are often not admissible in the context of constructing probability spaces.

We have shown that the sample space is explicitly constructible if we are given a prime p (i.e, as in a case we are given a set S and and integer $m < |S|$, and are asked to construct a sample space Ω for S^m, where $|S|$ is a prime). But what happens if we are not given p (i.e., as in case we are asked to construct such space when either $|S|$ is not a prime or $m > |S|$, and we proceed by constructing a sample space over a bigger S' of prime cardinality, which we have to find by ourselves)? In this case we face the problem of finding big primes (when the desired set is large). This is considered to be a difficult task (for deterministic algorithms). Typically, in such a case, it will be better to seek a field of cardinality p^e for some small prime p (e.g., 2) and big integer e. We'll consider this case in a while.

We should be able to represent explicitly elements of the field, that is trivial task in the case of a field of prime cardinality, and to make field operations, like addition and multiplication. Performing these operations is relatively easy when we are talking about a field of a prime cardinality (the operations are simply addition and multiplication modulo the prime).

Now we proceed to the case the cardinality of field is a prime power. We denote such a field as $GF(p^e)$, where p is a prime, and e is an integer bigger than 1. We again should give an answer on the following list of questions:

1. How do we represent elements of this field?
2. How do we operations in it, i.e. how can addition and multiplication be performed in such a field?

The set of elements of that kind of the field is typically represented by polynomials of degree smaller or equal to $e-1$ over $GF(p)$. This is the canonical representation of this kind of the field. Here we assume that we already know how to operate with $GF(p)$. In particular, in the case of $GF(2)$ it's very easy. We are talking about numbers 0 and 1, and operations are addition and multiplication modulo 2.

Having defined the set of the elements, now we should equipped the field with operations. In this representation the field always comes with an irreducible polynomial of degree exactly e over $GF(p)$. A polynomial is called irreducible, if it has no non-trivial polynomial division. In other words, $Q(x)$ is irreducible over $F(p)$ if for every two polynomials $Q_1(x)$ and $Q_2(x)$ over $GF(p)$ if

$$Q = Q_1 \cdot Q_2$$

then one of these polynomials is actually a constant.

Now we are ready to define the operations. Addition is easily implemented as an addition of polynomials. Remember, that elements of our field are polynomials, and therefore we may add elements in the field directly as polynomials. We obtain the description of the field operation addition:

`addition`: *simple addition of polynomials*

We would like to introduce multiplication in the field also as a multiplication of the polynomials. Unfortunately, we cannot do it so easily, because product of two polynomials of degree less or equal to $e-1$ may be the polynomial of degree bigger than $e-1$, which is not a valid representation of a field element. The right way is to define this operation is as multiplication modulo irreducible polynomial, denoted Q. It means that we do multiplication in the following way: firstly, we calculate usual product of two polynomials and then we reduce the result modulo the irreducible polynomial $Q(x)$. To clarify the notion of reduction modulo polynomial, let's write explicitly $Q(x)$. Without loss of generality, we may assume that in $Q(x)$ coefficient of x^e is 1:

$$Q(x) = x^e + \sum_{i=0}^{e-1} q_i \cdot x^i \quad (\text{in } GF(p))$$

Reduction modulo $Q(x)$ implies that the polynomial $Q(x)$ is considered to be a zero of the field $GF(p^e)$:

$$Q(x) = x^e + \sum_{i=0}^{e-1} q_i \cdot x^i = 0$$

Thus in $GF(p^e)$ we have

$$x^e = -\sum_{i=0}^{e-1} q_i \cdot x^i$$

Now whenever we encounter term x^e in result of product of polynomials, we may replace it by $-\sum_{i=0}^{e-1} q_i \cdot x^i$. Furthermore, we may replace x^{e+1} firstly by $x \cdot x^e$, and then substitute the above representation of x^e.

$$x^{e+1} = x \cdot x^e = x \cdot (-\sum_{i=0}^{e-1} q_i \cdot x^i) = -\sum_{i=0}^{e-1} q_i \cdot x^{i+1}$$

$$= -\sum_{i=0}^{e-2} q_i \cdot x^{i+1} - q_{e-1} \cdot x^e = -\sum_{i=0}^{e-2} q_i \cdot x^{i+1} + q_{e-1} \cdot \sum_{i=0}^{e-1} q_i \cdot x^i$$

So we have recursively represented x^{e+1} as a polynomial of degree less or equal to $e-1$. Obviously, we may represent any x^{e+k},(where $k = 1 ... e-2$) as a polynomial of degree less or equal to $e-1$ by the same recursive procedure.

Finally, we are able to represent any possible product of 2 polynomials degree less or equal to $e-1$ (it can be obviously a polynomial of degree up to $2e-2$) as a polynomial of degree less or equal to $e-1$. So now we complete our description of the field by introducing the operation of multiplication:

`multiplication`: *multiplication of polynomials modulo irreducible polynomial* $Q(x)$.

Unfortunately, there exists an number of difficulties. The problems are connected with the field representation of sample space. They don't appear when the set of field elements is somewhat small, because small primes can be found very easily. But if $|S|$ (or rather $|S'|$) has to be very big, then there are two alternative for field representation: either through a field of prime cardinality or through a field, whose size is a prime power. The former leads us to problem of finding big primes via a deterministic algorithm. Up to now it's not clear how we can solve it effectively. There exists probabilistic algorithms for big prime finding, but by definition it only gives a prime with *high probability*, while we would like to have a deterministic algorithm that produces a big prime.

The other option for the field representation; that is, as a field of prime power cardinality, seems to be less problematic. If p is small, then we just need to find an irreducible polynomial. There are solutions to this issue also, but they are beyond the scope of this lecture. We only give one special case for $GF(2^l)$. If l is of the form $l = 2 * 3^i$,where i is an integer, then the following polynomial is irreducible:

$$x^l + x^{l/2} + 1$$

Now, having resolved the technical difficulties of the construction, let's proceed to a nice application of this technique.

3 Application to 3CNF

Recall the algorithm for finding assignment for 3CNF formula that satisfies at least 7/8 clauses (presented on the first lecture). The algorithm works as follows: choose random assignment (uniformly) and test it. If it satisfies more than 7/8 of clauses then output the assignment, else choose another assignment. This algorithm works in expected polynomial time and requires a source of random bits. Here we present a modification that works in deterministic polynomial time.

Denote by m the number of variables in the input formula φ and by n the number of clauses. Let $\Omega \subset \{0,1\}^m$ be three-wise independent sample space. Using the technique described above, the size of Ω is of the order m^3. We claim that the expected number of clauses satisfied by a uniformly selected assignment in Ω equals $\frac{7}{8} \cdot n$ (exactly as is the case when selecting a totally random assignment (i.e., a uniformly selected assignment in $\{0,1\}^m$)):

Claim 3.1

$$\mathbf{E}_{\tau \in_R \Omega}(\text{number of clauses satisfied by } \tau) = \frac{7}{8} \cdot \text{number of clauses.}$$

Proof: Denote by I_i an indicator of the event that i'th clause is satisfied. Then,

$$\mathbf{E}_{\tau \in_R \Omega}[\text{number of clauses satisfied by } \tau] = \mathbf{E}(\sum_{i=1}^{n} I_i) = n \cdot \mathbf{E}(I_1),$$

where equality is due to linearity of expectation and to the fact that the I_j's are identically distributed. In order to compute $\mathbf{E}(I_1)$ recall that the distribution of individual letters in the assignment is three-wise independent and the fact that in each clause we have exactly three variables hence they are uniformly distributed over the set $\{0,1\}^3$. Thus,

$$\mathbf{E}(I_1) = \Pr[\text{clause is satisfied }] \cdot 1 + \Pr[\text{clause is not satisfied}] \cdot 0 = 7/8.$$

The claim follows. ■

Recall that $\Pr[X \geq E(X)] > 0$ for every random variable X. Letting X represent the number of clauses satisfied by an assignment uniformly selected in Ω, we conclude that Ω must contain an assignment that satisfies at least the expected number of satisfied clauses, (i.e., a 7/8th fraction of the clauses). Therefore, it is enough to scan the whole sample space Ω in search for such an assignment, and since the size of the space is only $O(m^3)$ we can do it efficiently.

4 Conclusion

The construction we have presented in this lecture is rather elegant. It enables dealing with k-wise independent elements in substantially small sample space. For the sake of simplicity, let's suppose that S is a field, and m is smaller than the size of S Thus we obtain k-wise independent sample space Ω and the size of the construction $|\Omega|$ is equal to $|S|^k$. This sample space can be explicitly constructed, and we have given an algorithm for it. One can write down all the elements of it in time:

$$O(m \cdot |S|^k) \cdot (time\ of\ field\ operations).$$

The size of our construction is (in some sense) optimal in terms $|S|$, because as we have discussed before, the size of construction of uniform over S variables with k-wise independency property cannot be less than $|S|^k$.

In case we are dealing with $S = \{0,1\}$ and we follow the recipe which was described in Section 2, then the size of the sample space will be m^k. Let's consider the optimality of this size. More exactly, whether we can have the construction with the same property, but of substantially smaller size than m^k. The answer is no. There is a lower bound for k-wise independent sample space of sequences of length m over $S = \{0,1\}$ that asserts that that the size of such a space must be at a least $m^{\lfloor k/2 \rfloor}$. That is, the basis is m and the exponent is k.