Instructions. Do this as part of your HW group, and submit one writeup per group. You can use your own class notes but no other resources including the Web.

Notation. We sometimes use the standard notation $[\ell]$ to denote the set \{1, 2, \ldots, \ell\}; “r.v.” is shorthand for “random variable”.

One of the strong indicators of the utility of “probabilistic thinking” in computer science is provided by the area of derandomization, which deals with the efficient transformation of probabilistic algorithms and arguments into efficient deterministic algorithms. For many natural problems, the current best deterministic algorithms start with a randomized approach and then apply a derandomization tool to efficiently “remove the randomness” from the algorithm. The success of this indirect approach to the most desirable type of algorithm—efficient and deterministic—points to the fruitfulness of probabilistic thinking.

1. Suppose there is some joint distribution $D$ on $n$ random variables $X_1, \ldots, X_n$, each taking values in $[m]$, such that
$$E_D[f(X_1, X_2, \ldots, X_n)] \geq a,$$
for some function $f$ and some value $a$. Call a point $\overline{x} = (x_1, x_2, \ldots, x_n) \in [m]^n$ good if $f(\overline{x}) \geq a$, and suppose we want to find any good point $\overline{g} = (g_1, g_2, \ldots, g_n)$. Clearly, some good point exists; how to find it efficiently and deterministically? Consider the following candidate algorithm:

for $i:= 1$ to $n$ do:
    breaking ties arbitrarily, set $g_i$ to be the index in $[m]$ such that
    $$E_D[f(\overline{X}) \mid X_i = g_i, \bigwedge_{j=1}^{i-1}(X_j = g_j)] \geq E_D[f(\overline{X}) \mid X_i = t, \bigwedge_{j=1}^{i-1}(X_j = g_j)], \forall t \in [m]$$
Thus, we deterministically generate $(g_1, g_2, \ldots, g_n)$. The key claim is that
$$E_D[f(\overline{X}) \mid \bigwedge_{j=1}^{i+1}(X_j = g_j)] \geq E_D[f(\overline{X}) \mid \bigwedge_{j=1}^{i}(X_j = g_j)], \ i = 0, \ldots, n-1,$$ (1)
which implies that
$$f(\overline{g}) = E_D[f(\overline{X}) \mid X_j = g_j, j = 1, 2, \ldots, n] \geq E_D[f(\overline{X})] \geq a,$$
i.e., that \( \bar{g} \) is good. Prove the claim (1).

2. Consider graphs \( G = (V,E) \), where \( m \) as usual is the number of edges, and \( d_v \) is the degree of vertex \( v \). Recal two randomized algorithms that we studied early on: (a) the simple randomized algorithm to construct a cut of size at least \( m/2 \), and (b) randomly permuting the vertices and running a simple algorithm at each step, in order to find an independent set of expected size \( \sum_{v \in V} 1/(d_v + 1) \). What are the algorithms you obtain when you apply the derandomization of problem 1 to each of these algorithms? These are simply-described deterministic algorithms: find what they are.

3. Often, it is not clear as to how to compute the relevant conditional probabilities/expectations such as those above exactly in an efficient manner. We will now study certain approximations to them are sufficient. For variety, we now describe this method in terms of a probability being smaller than 1, rather than an expectation being at least a certain quantity.

For our purposes, we focus on the case of independent binary r.v.s. Let \( X_1, X_2, \ldots, X_\ell \in \{0,1\} \) be independent r.v.s with \( \Pr(X_i = 1) = p_i \), for some \( p \in [0,1]^\ell \). Define the prefix-vector \( X^{(i)} \) to be \( (X_1, X_2, \ldots, X_i) \), for any \( i \in [\ell] \). Suppose, for some implicitly defined \( L \subseteq \{0,1\}^\ell \), that

\[
\Pr(X^{(i)} \in L) < 1.
\]

How do we efficiently and deterministically find some \( v \in \{0,1\}^\ell - L \)?

**Notation 1** \( \forall q \in [0,1]^\ell \) \( \forall i \in (\{0\} \cup [\ell]) \) \( \forall w \in \{0,1\}^i \), let

\[
u(i, w, q) = (w_1, w_2, \ldots, w_i, q_{i+1}, q_{i+2}, \ldots, q_{\ell}),
\]

and for any \( j \in \{0,1\} \), define \( w^j \in \{0,1\}^{i+1} \) as the string \( (w_1, w_2, \ldots, w_i, j) \).

Returning to the \( X_i \)s, \( p \) and \( L \), we define a function \( U : [0,1]^\ell \to \mathbb{R}^+ \) to be a good approximator w.r.t. \( (X_1, \ldots, X_\ell) \) and \( L \) if:

1. \( U(p_1, p_2, \ldots, p_\ell) < 1 \), and
2. \( \forall i \in \{0\} \cup [\ell] \) \( \forall w \in \{0,1\}^i \),
   (a) \( U(u(i, w, p)) \geq \Pr(X^{(i)} \in L | X^{(i)} = w) \), and
   (b) if \( i \leq \ell - 1 \), then \( U(u(i, w, p)) \geq \min\{U(u(i+1, w0, p)), U(u(i+1, w1, p))\} \).

Now let an efficiently computable \( U \) be a good approximator w.r.t. \( (X_1, \ldots, X_\ell) \) and \( L \). Defining, \( \forall i \in \{0\} \cup [\ell - 1] \) \( \forall w \in \{0,1\}^i \), \( n(i, w) = j \in \{0,1\} \) by

\[
U(u(i + 1, wj, p)) = \min\{U(u(i + 1, w0, p)), U(u(i + 1, w1, p))\}
\]

by breaking ties arbitrarily, prove that the following algorithm produces a \( v \notin L \):

For \( i := 0 \) to \( \ell - 1 \) do: \( v_{i+1} := n(i, v^{(i)}) \).
4. Although \(k\)-wise independence (for small \(k\)) is a powerful approach to derandomization as seen in class, it has its limitations: if we insist on polynomial-sized sample spaces, then we can only work with \(k = O(1)\), which is not sufficient in many situations. A key observation here is that it should often suffice to approximate a given distribution using a much smaller sample space, since probabilistic analyses may be expected to be robust under small perturbations of the probabilities. (By “sample space”, we just mean a (multi-)set of values over which our probability distribution is defined: for \(n\)-bit vectors, the natural sample space is \(\{0,1\}^n\), which we want to substantially reduce: i.e., the number of \(n\)-bit vectors with nonzero probability should be much smaller than \(2^n\).) Specifically, (3) below gives a good approximation to \(k\)-wise independence.

Define a sample space \(X\) for \(n\)-bit vectors, along with its associated probability distribution \(D\), to be \(k\)-wise \(\epsilon\)-bad if

\[
\forall S \subseteq [n] \text{ with } 1 \leq |S| \leq k, \quad |Pr(\bigoplus_{i \in S} x_i = 1) - Pr(\bigoplus_{i \in S} x_i = 0)| \leq \epsilon,
\]

(2)

where \(\bigoplus\) denotes the XOR function and \(x_i\) denotes the \(i\)th bit of an \(n\)-bit string \(x\) picked according to \(D\). In this problem, we will prove the following key “approximating \(k\)-wise independence” property of such a sample space:

\[
\forall \ell \in [k] \forall \{i_1, \ldots, i_\ell\} \subseteq [n] \forall b_1 b_2 \cdots b_\ell \in \{0,1\}^\ell, \quad |Pr(x_{i_1} = b_1, x_{i_2} = b_2, \ldots, x_{i_\ell} = b_\ell) - \frac{1}{2^\ell}| \leq \epsilon.
\]

(3)

Thus, it would suffice to demonstrate “small” samples spaces that satisfy (2).

Keep in mind that we want to show that (2) implies (3). We will see that this follows by appropriately “rotating the coordinate axes”. Order the elements of \(\{0,1\}^n\) in some arbitrary way as \(a_1, a_2, \ldots, a_{2^n}\). Consider the space of all functions that map \(\{0,1\}^n\) to the reals; define the (standard) dot-product of two such functions \(f\) and \(g\) as

\[
f \cdot g = \sum_{b \in \{0,1\}^n} f(b)g(b).
\]

The natural way to write any such a function \(f\) as a \(2^n\)-dimensional vector is just by listing all its \(2^n\) values, as the tuple

\[(f(a_1), f(a_2), \ldots, f(a_{2^n})).\]

That is, let \(e_i\) be the special function that maps element \(a_i\) to 1, and all other \(a_j\) to 0. Then, we are writing our function \(f\) as

\[f = \sum_{i=1}^{2^n} (f \cdot e_i) e_i,\]

(4)

i.e., writing \(f\) as a linear combination of the coordinate axes \(e_i\), where the component of \(f\) along \(e_i\) is the dot-product \(f \cdot e_i\). (Note that (4) just denotes that for all \(b \in \{0,1\}^n\), \(f(b) = \sum_{i=1}^{2^n} (f \cdot e_i) e_i(b)\).)
However, suppose we define an alternative set of coordinate axes: i.e., a collection of $2^n$ vectors, and a slightly different dot-product. Again consider the space of all functions that map $\{0,1\}^n$ to the reals; define the (normalized) dot-product of two such functions $f$ and $g$ as

$$\langle f, g \rangle = 2^{-n} \sum_{b \in \{0,1\}^n} f(b)g(b).$$

Our $2^n$ basis vectors (i.e., coordinate axes) now will each correspond to one subset $S$ of $[n]$; indeed, there are $2^n$ such subsets. The function $\chi_S$ corresponding to $S \subseteq [n]$ is defined by

$$\forall b = (b_1, b_2, \ldots, b_n) \in \{0,1\}^n, \chi_S(b) = (-1)^{\sum_{i \in S} b_i};$$

in particular, for the empty set $S = \emptyset$, we have $\chi_\emptyset(b) = 1$ for all $b$.

(i) Prove that these functions $\chi_S$ form an orthonormal basis: i.e., $\langle \chi_S, \chi_S \rangle = 1$ for all $S$, and $\langle \chi_S, \chi_T \rangle = 0$ if $S \neq T$.

(ii) We are now ready to prove that (2) implies (3). Suppose first for simplicity that $k = n$.\footnote{Prove for this case, and then show how your proof easily extends to the general case.} By part (i), any function $f$ mapping $\{0,1\}^n$ to the reals can be written as

$$f = \sum_{S \subseteq [n]} \langle f, \chi_S \rangle \chi_S,$$  \hspace{1cm} (5)

analogously to (4). Now, letting $f$ be the function

$$f(b_1, b_2, \ldots, b_n) = Pr(x_1 = b_1, x_2 = b_2, \ldots, x_n = b_n),$$

start with (5) and complete the proof.

**Remark:** It is known how to explicitly construct a multi-set $S$ of $n$-bit vectors such that the uniform distribution over $S$ satisfies (2), and such that $S$ is small enough: specifically,

$$|S| = O\left( \min \left\{ \frac{k \log n}{\epsilon^3}, \frac{k^2 \log^2 n}{\epsilon^2} \right\} \right).$$

\footnote{Prove for this case, and then show how your proof easily extends to the general case.}