CSE 525: Randomized Algorithms **Fall 2026**

Lecture 3: DNF Counting and Unreliability

Lecturer: Shayan Oveis Gharan *04/02/2026* *Scribe:*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

3.1 Recap from CSE 521: Unbiased Estimators

We say a random variable X is an unbiased estimator of μ if

$$\mathbb{E}[X] = \mu.$$

It turns out the the number of samples is proportional to the relative variance of X.

Definition 3.1 (Relative Variance)**.** *Say X is an unbiased estimator of μ, then, the relative variance of X is defined as*

$$\frac{\sigma^2(X)}{\mu^2}, \tag{3.1}$$

where by $\sigma^2(X) = \mathbb{E}[X]^2 - (\mathbb{E}[X])^2$ is the variance of X. We typically use t to denote the relative variance.

The following theorem is the main result of this section.

Theorem 3.2. *Given $\epsilon, \delta > 0$, and an unbiased estimator of μ, X. We can approximate μ within $1 \pm \epsilon$ multiplicative factor using only $O(\frac{t}{\epsilon^2} \log \frac{1}{\delta})$ independent samples of X with probability $1 - \delta$.*

Monte Carlo Simulation. Suppose we want to estimate the size of a set S. There is a generic approach called the Monte Carlo simulation. First, we find a univere $U \supseteq S$ that has the following two properties:

- We know the size of U
- We can efficiently sample from U.

We sample an element randomly from U and test whether it belongs to S. Let X be the corresponding indicator random variable. It follows that $Y = |X| \cdot |U|$ is an unbiased estimator of S with relative variance, $\frac{|U|}{|S|}$. So, we can estimate $|S|$ within a $1 \pm \epsilon$ multiplicative factor by generating $O(\frac{|U|}{\epsilon^2 |S|} \log \frac{1}{\delta})$ many samples from U.

3.2 FPRAS for DNF Counting

A DNF is conjunction of clauses where each of them is a disjunction of literals, e.g., $(x_1 \wedge \overline{x_2} \wedge x_4) \vee (\overline{x_3} \wedge x_2) \ldots$. Consider a DNF with n variables. Obviously the problem of finding a satisfying assignment for a DNF is in P. We want to count the number of satisfying assignment to a DNF. Here we prove the following theorem:

Theorem 3.3 ([**KLM89**])**.** *There is an FPRAS for the DNF counting problem.*

Following the Monte Carlo sampling method, we can let S be the set of satisfying assignments to the given DNF. If U is the set of all truth assignments to the n variables then it has the desired properties. So, by the discussion in the previous section we would need to generate $O(\frac{2^n}{\epsilon^2|S|})$ many samples. So, this would work only if $|S|$ is within a polynomial factor of 2^n; but what if it is exponentially smaller?

The ida of Karp, Luby and Madras [**KLM89**] is to choose the univese U carefully such that $|S|$ is within a polynomial factor of $|U|$. In fact, they consider a more general problem. Suppose we have sets $S_1, S_2, \dots, S_m$ of a ground set of elements and we want to estimate $|\cup_i S_i|$. First, of all observe that we always have

$$\frac{1}{m} \leq \frac{|\cup_i S_i|}{\sum_i |S_i|} \leq 1$$

So, the idea is to construct an artificial universe U of size $|U| = \sum_i |S_i|$. How can we do that? It is enough that for each ground element e, and for every set S_i where $e \in S_i$ put a distinct copy of e in U:

$$U := \{(e, i) : e \in S_i\}.$$

By definition $|U| = \sum_i |S_i|$.

Now suppose we know the sizes of all S_i's and we can sample a u.r. element from each efficiently. Then, we claim that U is an ideal universe:

- We can compute the size of $|U| = \sum_i |S_i|$ exactly.
- To sample an element uniformly at random from U, we first sample i with probability $\frac{|S_i|}{|U|}$. Then, we sample a uniformly random element from S_i.

Now, let $S = \cup_i S_i$ and notice that $U \not\supset S$. So, we need identify some of the elements of U with S; in particular, for each element $e \in S$, let i be the smallest index such that $e \in S_i$; we identify $e \in S$ with (e, i). So, to run the the Monte Carlo method, we first sample an element (e, i) from U and then we check whether $(e, i) \in S$ by simply checking wether i is the smallest index where S_i contains e.

In our DNF counting problem each S_i corresponds to the set of assignments that satisfy the i-th clause. Obviously if the i-th clause has k literals, $|S_i| = 2^{n-k}$, and we can sample efficiently from S_i simply by fixing every literal that appears in S_i to be true and choosing rest uniformly at random.

3.3 Network Unreliability

As an application we discuss an algorithm for the network unreliability problem. Given a network of n vertices where each edge e disappears independently with probability p_e determine the probability that the surviving network is disconnected.

In this lecture for simplicity we assume that all p_e's are equal to p and we define $Fail(p)$ to denote the failure probability of the whole network. We prove the following theorem of Karger [**Kar95**]. Also, see [**Kar16**] for a much faster algorithm.

Theorem 3.4 ([**Kar95**])**.** *There is an FPRAS for the network reliability problem. That is given G, p and ϵ the algorithm returns a $1 + \epsilon$ multiplicative approximation to $Fail(p)$ with probability $3/4$.*

Note that the success probability can be easily boosted up to $1-\delta$ by running $O(\log(1/\delta))$ independent copies of the algorithm and returning the median of the estimates.

First, let us discuss how the two problems are related. Suppose our graph have r cuts, $C_1, \dots, C_r$. Then, we can use the above algorithm to estimate the probability that all of the edges in one of these cuts fail. In particular, we write

$$\vee_{1\le i\le r} \wedge_{e\in C_i} X_e$$

where X_e is the indicator (random variable) that edge e fails. Then, every realization of failures that makes G disconnected correspond to a satisfying assignment for the above DNF formula. So, we can use our FPRAS to get a $1+\epsilon$ approximation to the probability that at least one of the cuts $C_1, \dots, C_r$ fail.

Remark 3.5. *Note that there is a technical problem here, because each edge fails with probability p whereas in the proof of Theorem 3.3 we assumed that every variable is true/false with 1/2 probability. We leave this as an exercise.*

By the above discussion if G has polynomially many cuts then we are already done. The problem is that G has exponentially many cuts; i.e., we need to find the union of exponentially many sets. Karger's idea is as follows: Let k denote the size of the minimum cut of G. Then, obviously,

$$Fail(p) \ge p^k =: q. \tag{3.2}$$

Now, we consider two cases

Case 1: $q > 1/\operatorname{poly}(n)$ In this case, we can simply use Monte Carlo method to obtain a $1+\epsilon$ approximation of $Fail(p)$ using only $O(\frac{1}{\epsilon^2 q}\log\frac{1}{\delta})$ many samples. In other words, we let U be the universe of all realizations of G.

Case 2: $q < 1/\operatorname{poly}(n)$ As it will be clear we need to let $\operatorname{poly}(n)$ be n^4. Here the idea is to divide all cuts of G into two groups: (i) Near minimum cuts and (ii) Large cuts. We use the following theorem of Karger to prove that there are only polynomialy many near minimum cut in any graph G. So, we can use Theorem 3.3 to estimate the probability that at least one of these cuts fails. To deal with large cuts we also use the following theorem to argue that it is very unlikely that any of the large cuts fails.

Theorem 3.6 (Karger [**Kar95**])**.** *For any graph G with n vertices and with minimum cut k, and for any $\alpha \ge 1$, the number of cuts of size at most αk in is at most $n^{2\alpha}$.*

The above theorem can be proved by Karger's contraction algorithm, see 521-Notes. Also, note that the statement of the above theorem is tight, as in a cycle of length n there are $O(n^{2\alpha})$ cuts with 2α edges.

Now, let α be a parameter that we fix later. We want to show that with probability at most ϵq all cuts of size at least αk survive.

Lemma 3.7. *Let $C_1, \dots, C_r$ be all cuts of G and let us sort them in the order of their size*

$$|C_1| \le |C_2| \le \dots \le |C_r|.$$

For any $\alpha \ge 1$, and $q = n^{-\beta}$ we have

$$\mathbb{P}\left[\exists i \ge n^{2\alpha} : C_i \text{ fails}\right] \le \frac{n^{2\alpha(-\beta/2+1)}}{\beta/2-1}$$

Proof. Firstly, by Theorem 3.6, for any $i = n^{2x}$, we have $|C_i| \geq xk$. In other words,

$$|C_i| \geq \frac{\log i}{2 \log n} k \tag{3.3}$$

By union bound can write

$$\begin{aligned} \mathbb{P}\left[\exists i \geq n^{2\alpha} : C_i \text{ fails}\right] &\leq \sum_{i \geq n^{2\alpha}} \mathbb{P}\left[C_i \text{ fails}\right] \\ &= \sum_{i \geq n^{2\alpha}} p^{|C_i|} \\ &\underset{(3.3)}{\leq} \sum_{i \geq n^{2\alpha}} p^{\frac{\log i}{2 \log n} k} \\ &\underset{(3.2)}{=} \sum_{i \geq n^{2\alpha}} q^{\log(i)/2\log(n)}. \end{aligned}$$

Say $q = n^{-\beta}$. We can write $q^{\log(i)/2\log(n)} = e^{-\beta \log(i)/2} = i^{-\beta/2}$. Therefore,

$$\mathbb{P}\left[\exists i \geq n^{2\alpha} : C_i \text{ fails}\right] \leq \int_{x \geq n^{2\alpha}} x^{-\beta/2} dx = \frac{x^{-\beta/2+1}}{-\beta/2+1}\Big|_{n^{2\alpha}}^{\infty} = \frac{n^{2\alpha(-\beta/2+1)}}{\beta/2-1}$$

□

So, to make sure that all large cuts survive it is enough to choose α such that the above probability is at most ϵq. Observe that if we choose $\alpha = 2$ and making sure that $q \leq \epsilon \cdot n^{-4}$ we get $\beta \geq 4$, so

$$\mathbb{P}\left[\exists i \geq n^{2\alpha} : C_i \text{ fails}\right] \leq n^{-\beta\alpha+2\alpha} \underset{\alpha=2}{=} n^{-2\beta+4} \underset{q=n^{-\beta}}{=} q^2 \cdot n^4 \underset{q \leq \epsilon n^{-4}}{\leq} \epsilon q.$$

So, it is enough to solve the DNF counting problem for $n^{2\alpha} = n^4$ smallest cuts. These cuts can be found in polynomial time using Karger's contraction algorithm.