Lecture 12: DNF Counting, Unreliability Estimation, #P

October 3, 2025

1 Basic Estimation Framework

Suppose we have a finite universe U of N elements and $S \subseteq U$ of n elements. We know N and can sample uniformly at random from U. We want to estimate $|S|$.

We also assume that given $x \in U$, we can efficiently check if $x \in S$.

Basic Algorithm: We can estimate $|S|$ by taking a sample $X \in U$ and outputting N if $X \in S$ and 0 otherwise. It is easy to see that this is an exact estimator for $|S|$, but variance is $\Theta(N^2)$.

Thus, using our standard tricks, we can obtain an (ϵ, δ) approximation of $|S|$ using $O\left(\frac{1}{\epsilon^2}\right)$ samples.

Therefore, if n is not too large and we can sample from U uniformly, we can estimate $|S|$. The goal is to see two nice applications of this simple idea and also introduce briefly the counting complexity class #P defined by Valiant in his influential work.

Note: We can apply the above estimator even in the continuous setting where we have a probability measure μ on U and we want to estimate $\mu(S)$.

2 #P and DNF Counting

A **DNF formula** over n boolean variables $X_1, X_2, \ldots, X_n$ is a formula

$$\phi = C_1 \vee C_2 \vee \cdots \vee C_m$$

which is a disjunction (OR) of several clauses, each of which is a conjunction of a set of literals.

Example:

$$(X_1 \wedge \bar{X}_3 \wedge X_5) \vee (X_2 \wedge \bar{X}_4 \wedge X_1 \wedge X_6)$$

Clearly any DNF formula is satisfiable. We want to count the number of satisfying assignments to ϕ. We will denote it by $|\phi|$. Exact counting is likely to be hard since it is Complete for the counting class #P.

However, we can get an (ϵ, δ) approximation in poly $\left(m, n, \frac{1}{\epsilon}, \log\frac{1}{\delta}\right)$ time.

2.1 Naive Approach

Let U be the set of 2^n possible Boolean assignments to the variables. Let S be those that satisfy ϕ.

We can sample from U and it is easy to check if $X \in S$. Hence we can apply the basic scheme. Unfortunately, $\frac{|S|}{|U|}$ can be exponentially small. Hence this naive scheme does not work.

2.2 Refined Idea

Let S_i be the set of all assignments that satisfy clause C_i.

We know $|S_i|$ and can also generate a uniform sample from S_i easily.

How? If $C_i = X_1 \wedge \bar{X}_2 \wedge X_5$, then an assignment to the n variables satisfies C_i iff $X_1 = 1$, $X_2 = 0$, $X_5 = 1$. Thus $|S_i| = 2^{n-3}$. We can sample from S_i uniformly as well.

We want to estimate $|S| = |S_1 \cup S_2 \cup \cdots \cup S_m|$. A satisfying truth assignment may appear in multiple sets.

To use our basic set up of U and S, we do the following:

i. Let $\beta \in S$ be a satisfying truth assignment. We let

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

In other words, we create a copy of β for each S_i it belongs to.

ii. We map (β, i) to S if i is the smallest index such that $\beta \in S_i$.

Now $|U| = \sum_i |S_i|$ and $|S| = |\phi|$, and thus

$$\frac{|S|}{|U|} = \frac{|\phi|}{\sum_i |S_i|} \geq \frac{1}{m}.$$

Sampling from U: We pick $i \in [m]$ where probability of i appearing is $\frac{|S_i|}{|U|}$. Once i is picked, we pick a uniformly random β from S_i. It is easy to see that this process generates a uniform element from U.

Checking membership in S: Given (β, i), we check if i is the smallest index such that $\beta \in S_i$. If it is, then we output it as an element from S.

Thus we can estimate $|\phi|$ to within a $(1 \pm \epsilon)$ factor with high probability using $O\left(\frac{m}{\epsilon^2}\right)$ samples. Each sample can be generated and checked in poly time.

2.3 A Natural Example

Given shapes, want to estimate area of their union. If shapes are well behaved, can do exact computation but expensive. Can obtain fast approximation generically. Only need to have information for each shape.

3 Unreliability of a Graph

Given undirected graph $G = (V, E)$ and for each edge $e \in E$ a value $p_e \in [0, 1]$, where p_e is the probability that e fails.

Suppose each edge fails independently with probability p_e.

Let α be the probability that G is disconnected. This is the **unreliability estimation problem**.

We can also be interested in estimating $1 - \alpha = \beta$. Note that approximating α is not same as approximating β.

We will discuss estimating α.

3.1 Two Regimes of Interest

(i) Easy Case: $\alpha \geq \frac{1}{c}$ for some sufficiently large constant c.

This is easy because we can run the Monte Carlo Simulation with $O\left(\frac{1}{\epsilon^2} \log \frac{1}{\delta}\right)$ experiments. We can estimate α to within $1 \pm \epsilon$ with high probability.

(ii) Difficult Case: When $\alpha \ll \frac{1}{c}$. This is when α is really small. Plain simulation will not work since we will rarely if ever see G being disconnected.

We will address the above case in the rest of the lecture.

3.2 Analysis for Small α

For simplicity, we will assume $p_e = p$ for all e. One can reduce the general case to this but requires some work.

Let K be the size of the min cut of G in terms of number of edges.

Let $C_1, C_2, \ldots, C_h$ be the cuts of G ordered in non-decreasing value. We treat each C_i as a set of edges.

Let A_i be the event that cut C_i fails. That is, all edges in C_i fail: $\prod_{e \in C_i} X_e$ where $X_e = 1$ with prob p. Then $\Pr[A_i] = p^{|C_i|}$.

G is disconnected iff $\bigcup_i A_i$.

If h was small (polynomially bounded) we could use previous ideas, but there are an exponential number of cuts.

Since $p_e = p$ for all e and $K \leq |C_i|$, we have $\Pr[A_i] \leq \Pr[A_j]$ if $i \leq j$.

3.3 Main Observation

We are in the setting that p is quite small (failure prob).

Also $\alpha = \Pr\left[\bigcup_i A_i\right] \geq \Pr[A_1] = p^K$.

Consider any cut C_j such that $|C_j| \geq 4K$. Then

$$\Pr[A_j] = p^{|C_j|} \leq p^{4K} = (p^K)^4.$$

So large cuts are very unlikely to fail. But there are a lot of large cuts.

However, as we saw earlier, the number of β-approximate cuts is only growing as $n^{2\beta}$.

Thus we can use union bound over all large cuts and still the total probability of them failing will be tiny compared to $\Pr[A_1]$. So we can ignore all cuts j such that $|C_j| \geq 4K$ if we choose c appropriately.

This implies we can focus on cuts j such that $K \leq |C_j| \leq 4K$, but there are only $O(n^8)$ of those cuts and we can run the basic estimation algorithm on these polynomially many cuts. Karger's algorithm can be used to enumerate all these cuts efficiently with high probability.

3.4 Formal Analysis

Lemma 1. $\displaystyle\sum_{j:|C_j|\geq cK} \Pr[A_j] \leq \frac{\alpha}{100}$

Proof. Consider all cuts j such that $K \leq |C_j| \leq cK$ for $c \geq 2$. Let l_i be the number of such cuts with $|C_j| = i$.

By Karger's theorem, $l_i \leq n^{2i/K}$.

Also $\Pr[A_j] = p^i$ for such C_j.

$$\begin{aligned}
\sum_{j:|C_j|\geq cK} \Pr[A_j] &\leq \sum_{i=cK}^{|E|} l_i \cdot p^i \\
&\leq \sum_{i=cK}^{|E|} n^{2i/K} \cdot p^i \\
&= \sum_{i=cK}^{|E|} \left(n^{2/K} \cdot p\right)^i \\
&\leq \frac{\left(n^{2/K} \cdot p\right)^{cK}}{1 - n^{2/K} \cdot p} \\
&\leq \frac{\alpha}{100}
\end{aligned}$$

for sufficiently large c. □

Thus we can ignore all large cuts and do the estimation with the polynomial number of small cuts.

Note: The above analysis is quite loose. See cited notes for better calculations.