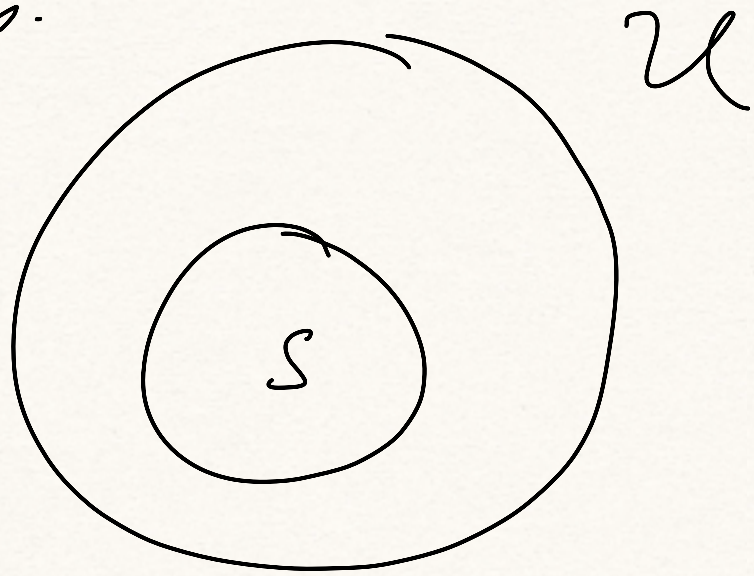
Lecture 12 10/3/2025

DNF Counting & Unreliability

Suppose we have a finite universe $\mathcal{U}$ of N elements and $S \subseteq N$ of n elements.



We know $|\mathcal{U}| = N$ and can sample uniformly at random from $\mathcal{U}$. Want to estimate $|S|$.
We also assume that given $x \in \mathcal{U}$ we can efficiently check if $x \in S$.

Then we can estimate $|S|$ by taking a sample $X \sim \mathcal{U}$ 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 $\left(\frac{|S|}{|\mathcal{U}|}\right)^2$.

Thus using our standard tricks we can obtain an (ε, δ) approximation for $|S|$ using $O\left(\frac{|\mathcal{U}|}{|S|} \cdot \frac{1}{\varepsilon^2} \ln \frac{1}{\delta}\right)$ samples.

Thus if $\frac{|\mathcal{U}|}{|S|}$ is not too large and we can sample from $\mathcal{U}$ 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 that we can apply above estimator even in the continuous setting where we have a probability measure μ on $\mathcal{U}$ and we want to estimate $\mu(S)$.

DNF Counting

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

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

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

$$\psi = X_1 X_2 X_5 + X_3 X_4 + \bar{X}_1 X_2 \bar{X}_5 \bar{X}_4$$

Clearly every DNF formula is satisfiable.

We want to count the number of satisfying assignments to ψ. We will denote it by $\#\psi$. Exact counting is likely to be hard since it is complete for the counting class $\#P$.

More on this later.

However we can get an (ε, δ) approximation in $\text{poly}(m,n) \frac{1}{\varepsilon^2} \ln \frac{1}{\delta}$ time.

<u>Basic</u>/<u>Naive</u> app<u>roach</u>: Let $\mathcal{U}$ be the set of all 2^n possible Boolean assignments to the variables. Let S be those that satisfy Ψ.

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

A 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 $C_i = x_1 x_2 \bar{x}_5$ then an assignment to the n variables satisfies C_i iff $x_1 = 1$, $x_2 = 1$, $x_5 = 0$

Thus $|S_i| = 2^{n-3}$. We can sample from $|S_i|$ uniformly as well.

We want to estimate $S = S_1 \cup S_2 \cdots \cup S_m$.

A satisfying truth assignment may appear in multiple sets.

To use our basic set up of $\mathcal{U}$ and S we do the following.

Let $\beta \in S$ be a satisfying truth assignment

We let $\mathcal{U} = \{ (\beta, i) \mid \beta \in S_i \}$

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

We map β to the smallest index $\underline{i}$ such that $(\beta, i) \in \mathcal{U}$.

Now $|\mathcal{U}| = \sum_{i=1}^{m} |S_i|$

and $|S| \leq |\mathcal{U}|$ and

$$|S| \geq \frac{|\mathcal{U}|}{m}$$

thus $$\frac{|S|}{|\mathcal{U}|} \leq m.$$

(i) We can sample uniformly from $\mathcal{U}$. How? We pick $i \in [m]$ where probability of i appearing is

$$\frac{|S_i|}{|\mathcal{U}|}.$$

Once i is picked we pick a uniformly random β from S_i.

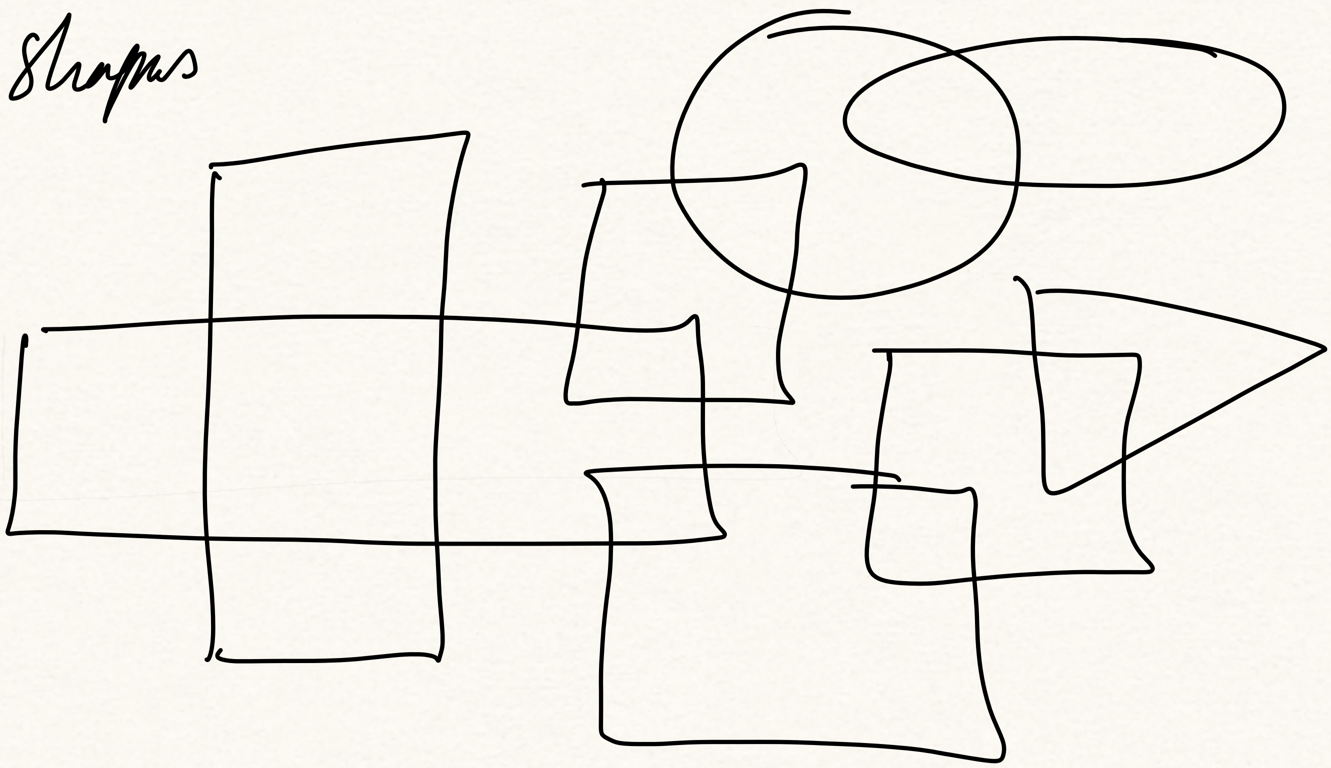
It is easy to see that this process generates a uniform element from $\mathcal{U}$.

(ii) 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 $\#\Psi$ to within a $(1 \pm \varepsilon)$ factor with high probability using $O\left(\frac{m \log n}{\varepsilon^2}\right)$ samples. Each sample can be generated and checked in poly time.

Another example:

given shapes



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.

Unreliability of a graph

Given undirected graph $G=(V,E)$ and for each edge $e \in E$ a value $p_e \in (0,1)$. 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.

Two regimes of interest

(i) $\alpha \geq \frac{1}{n^c}$ for some sufficiently large constant c.

This is easy case because we can run the Monte Carlo simulation with $O(\frac{1}{\alpha} \cdot \frac{1}{\varepsilon^2} \log n)$ experiments we can estimate α to within $(1+\varepsilon)$ with high probability.

(ii) The more difficult case is when $\alpha < \frac{1}{n^c}$. 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 above case in the rest of the lecture.

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 A_i fails. That is all edges in C_i fail $= \bigwedge_{e \in C_i} X_e$ where $X_e = 1$ with prob p. $\Rightarrow \Pr[A_i] = p^{|C_i|}$.

We are interested in $\bigcup_{i=1}^{h} A_i$.

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

Since $p_e = p$ $\forall e$ and $|C_i| \leq |C_j|$ we have $\Pr[A_i] \geq \Pr[A_j]$ $\forall i \leq j$.

Main observation is the following.

We are in the setting that

$\alpha \leq \frac{\varepsilon}{n^c}$ so quite small failure prob.

Also $\alpha \geq \Pr[A_1]$

$\Rightarrow \Pr[A_1] \leq \alpha$

$\Rightarrow p^k \leq \frac{\varepsilon}{n^c}.$

Consider any cut C_j such that

$|C_j| > 4k$

$$\Pr[A_j] < p^{4k} \leq \left(\frac{\varepsilon}{n^c}\right)^4 \leq \frac{\varepsilon^4}{n^{4c}}$$

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 $|C_j| > 4k$ if we choose c appropriately.

Implies we can focus on cuts j s.t. $|C_j| \leq 4k$ but there are only 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.

More formally

<u>Lemma:</u> $\sum_{j : |C_j| \geq 4k} \Pr[A_j] \leq$

<u>Proof:</u> Consider all cuts C_j

s.t. $|C_j| \in (2^i k, 2^{i+1} k]$ $i \geq 2$

Let l_i be the # of such cuts

$l_i \leq n^{2^{i+2}}$ by Karger's theorem

Also $\Pr[A_j] \leq p^{2^i k}$ $\forall$ such C_j

$$\Rightarrow \sum_{j : |C_j| \geq 4k} \Pr[A_j] \leq \sum_{i \geq 2} n^{2^{i+2}k} \cdot p^{2^i k}$$

$$\leq \sum_{i \geq 2} n^{2^{i+2}k} \left(\frac{\varepsilon}{n^c}\right)^{2^i}$$

$$\leq \sum_{i \geq 2} n^{(2^{i+2} - c \cdot 2^i)k} \cdot \varepsilon^{2^i}$$

$$\leq \frac{\varepsilon^2}{n^k} \leq \varepsilon \alpha.$$

for sufficiently large c.

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

$\square$

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