Probability and Algorithms **Caltech CS150, Winter 2003**
Leonard J. Schulman **Scribe: Sidharth Jaggi**
Notes for lecture 3, January 22, 2003. Overview; approximation of #DNF

Overview of the previous lecture:- A *language* is a decision problem. DNF is the language of satisfiable formulas in Disjunctive Normal Form. Thus formulas $\phi(x_1, \ldots, x_n)$ in DNF are of the form

$$\phi(x_1, \ldots, x_n) = \bigvee_{i=1}^{k} \left(\bigwedge_{j=1}^{r_i} l_{ij} \right)$$

where every l_{ij} is one of $x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n$, where the x_is are Boolean variables. Note that the computational complexity of satisfiability for DNF is trivial. This is because unless all the individual conjunctive clauses are individually non-satisfiable (which occurs for clause C_i if it contains both x_j and $\bar{x}_j$ for some j), one can select values for the x_i's such that at least one clause evaluates to 1. However, the problem of determining the existence of $\{x_i\}_{i=1}^n$ such that $\phi(x_1, \ldots, x_n) = 0$ is difficult.

This is the complement of the Conjunctive Normal Form (CNF), where the satisfiability problem is difficult, but determining the existence of $\{x_i\}_{i=1}^n$ such that $\phi(x_1, \ldots, x_n) = 0$ is trivial. DNF and CNF can be related to each other by DeMorgan's identities.

The function $\#DNF : \phi \to \mathbb{N}$ is defined for formulas ϕ in DNF to be the number of inputs $x = \{x_i\}_{i=1}^n$ satisfying ϕ.

$\#DNF \in \#P$, where $\#P$ is the set of functions $\{0,1\}^n \to \mathbb{N}$ which which give the number of satisfying leaves in the tree of a nondeterministic poly-time Turing machine. $\#DNF$ can be shown to be $\#P$-complete with respect to deterministic poly-time reductions.

Additive vs. Multiplicative Approximation

Let θ be the fraction of assignments satisfying ϕ. Therefore $\theta = |\{x : \phi(x) = 1\}|/2^n = \#DNF(\phi)/2^n$.
An *additive approximation* T satisfies $|T - \theta| < \epsilon$.
A *multiplicative approximation* T satisfies $\theta(1 - \epsilon) < T < \theta(1 + \epsilon)$.
Note that a multiplicative approximation implies an additive approximation, but not vice versa.
Eg:- Let $\phi(x) = (x_1 \wedge \ldots \wedge x_{n/3}) \vee (x_{n/3+1} \wedge \ldots \wedge x_{2n/3}) \vee (x_{2n/3+1} \wedge \ldots \wedge x_n)$. Then $\theta \approx 2^{-n/3}$. An additive approximation of no more than $1/\text{poly}(n)$ is useless as a multiplicative approximation.

Approximation algorithm (Karp, Luby and Madras)

(This will later be useful for network reliability problems, in particular the probability that a network remains connected when links undergo failure with some probability).

Let $\phi = C_1 \vee C_2 \vee \ldots \vee C_k$, where the C_i's are the clauses. For an assignment τ of x, let $\mu(\tau)$ be the number of clauses that τ satisfies. Let r_i be the number of literals in clause C_i. Define $q = \sum_i 2^{-r_i}$. If k is the number of clauses, then $\theta \leq q \leq k\theta$.

Algorithm:-

1. Choose a clause C_i with probability $2^{-r_i}/q$.
2. Uniformly and randomly, choose τ satisfying C_i.
3. Let T, our estimator of θ, be $T = q/\mu(\tau)$

Claim:- T is an unbiased estimator of θ. (That is, $E(T) = \theta$.)
Proof:- Examine the following set of equalities.

$$\begin{aligned} E\left(\frac{q}{\mu(\tau)}\right) &= \sum_i \frac{2^{-r_i}}{q} E\left(\frac{q}{\mu(\tau)} \mid \text{pick } C_i\right) \\ &= \sum_i \frac{2^{-r_i}}{q} \sum_{\tau: C_i(\tau)=1} 2^{r_i - n} \frac{q}{\mu(\tau)} \\ &= \sum_i \sum_{\tau: C_i(\tau)=1} \frac{2^{-n}}{\mu(\tau)} \\ &= 2^{-n} \sum_{\tau:\phi(\tau)=1} \frac{2^{-n}}{\mu(\tau)} \sum_{\tau: C_i(\tau)=1} 1 \\ &= 2^{-n} \sum_{\tau:\phi(\tau)=1} 1 \\ &= \theta. \end{aligned}$$

Note that even though the estimator obtained by sampling τ uniformly and checking for satisfiability is unbiased, the variance is huge. The estimator T obtained above, however, doesn't suffer from this problem, as shown below.

Variance of T:-

$$Var(T) = E(T - \theta)^2 = E(T^2) - \theta^2$$

But

$$\begin{aligned} E(T^2) &= \sum_i \frac{2^{-r_i}}{q} \sum_{\tau: C_i(\tau)=1} 2^{r_i - n} \frac{q^2}{\mu^2(\tau)} \\ &= 2^{-n} q \sum_i \sum_{\tau: C_i(\tau)=1} \frac{1}{\mu(\tau)} \\ &= 2^{-n} q \sum_{\tau:\phi(\tau)=1} \frac{1}{\mu(\tau)} \end{aligned} \tag{1}$$

Let $r = \min_i r_i$, therefore $\theta \geq 2^{-r}$. By the union bound, $\theta \leq k2^{-r}$, therefore $q \leq k2^{-r}$, which implies that

$$q \leq k\theta. \tag{2}$$

Also, since $\mu(\tau) \geq 1$, therefore

$$\sum_{\tau:\phi(\tau)=1} \frac{1}{\mu(\tau)} \leq 2^{-n}\theta \tag{3}$$

Substituting (2) and (3) in (1) gives us that $E(T^2) \leq k\theta^2$, and therefore

$$Var(T) \leq (k-1)\theta^2. \tag{4}$$

Amplification:-
The resulting variance is reduced in two steps, the second of which is shown in the next lecture.
1. Repeat the K-L-M procedure $\frac{k-1}{\epsilon^2\delta}$ times to get estimates $T_1, \ldots, T_{k-1/\epsilon^2\delta}$ of θ. Evaluate $\bar{T} = avg(T_i)$. Since the T_i are independent real random variables, therefore $Var(\bar{T}) = \frac{\epsilon^2\delta}{k-1} Var(T_i) = \frac{\epsilon^2\delta}{k-1}(k-1)\theta^2 = \epsilon^2\delta\theta^2$.

Interlude:-
Markov Inequality

If A is a non-negative random variable, then $\Pr(A \geq \epsilon) \leq \frac{E(A)}{\epsilon}$. This is because $E(A) = E(A|A < \epsilon)\Pr(A < \epsilon) + E(A|A \geq \epsilon)\Pr(A \geq \epsilon) \geq E(A|A \geq \epsilon)\Pr(A \geq \epsilon) \geq \epsilon \Pr(A \geq \epsilon)$.

Chebyshev Inequality
If $E(T) = \theta$, then $\Pr(|T - \theta| \geq c\sqrt{Var(T)} \leq \frac{1}{c^2}$.
Proof:- Apply Markov inequality to the random variable $(T - \theta)^2$.

Consequence for amplification of the algorithm:-
For $\bar{T}$ given above and $\theta = 2^{-n}\#DNF(\phi)$, $P(|\bar{T} - \theta| \geq \epsilon\theta) \leq \delta$.

Next time we'll do another amplification step and show how the K-L-M algorithm gives a "fully polynomial randomised approximation scheme" (FPRAS) for #DNF.