CSE5401 Lec.10

The counting-sampling connection

1. Downward self-reductions
2. Counting reduces to sampling
3. Sampling reduces to counting

Many of the problems we consider can be efficiently reduced to smaller instances of the same problem. For example, satisfiability: let $\varphi(x_1,\dots,x_n)$ be a CNF formula. Suppose we had an algorithm A_{n-1} that solved satisfiability on formulas with at most $n-1$ variables. Now, we can solve satisfiability on n variables:

On input $\varphi(x_1,\dots,x_n)$

Let $\varphi_0 \leftarrow \varphi(x_1,\dots,x_{n-1},0)$ (i.e., "plug in 0" - delete satisfied clauses & falsified literals)
If $A_{n-1}(\varphi_0) = (b_1,\dots,b_{n-1})(\neq \perp)$
  Return $(b_1,\dots,b_{n-1},0)$
Let $\varphi_1 \leftarrow \varphi(x_1,\dots,x_{n-1},1)$
If $A_{n-1}(\varphi_1) = (b_1,\dots,b_{n-1})(\neq \perp)$
  Return $(b_1,\dots,b_{n-1},1)$
Return $\perp$ (unsatisfiable)

Computing φ_0 & φ_1 is linear time — we just scan φ and copy only the remaining clauses & literals. So this is a linear-time algorithm. It is correct because if φ is satisfiable, then either there is a satisfying assignment that sets $x_n \leftarrow 0$, or one that sets $x_n \leftarrow 1$. In the former

case, φ_0 is satisfiable, and in the latter, φ_1 is. So A_{n-1} must return an assignment $(b_1,\dots,b_{n-1})$ in one of the two cases, where if φ_b was satisfied, $\varphi(b_1,\dots,b_{n-1},b)=\varphi_b(b_1,\dots,b_{n-1})$, so we indeed return a satisfying assignment to φ. Conversely, if one of the φ_b returns $(b_1,\dots,b_{n-1})$, indeed φ was satisfiable and $(b_1,\dots,b_{n-1},b)$ is a satisfying assignment.

We can abstract this as follows:

Defn: The relation problem $R \subseteq X \times Y$ has a downwards self reduction if

① $\exists$ poly-time function $m: X \to \mathbb{N}$ s.t. if $(x,y) \in R$, $|y| = m(x)$ (in bits)

② $\exists$ poly-time functions $\sigma: X \to \mathbb{N}$, $\psi: X \times Y \to X$ s.t.

$\sigma(x) \leq O(\log|x|)$; if $m(x) > 0$ then $\sigma(x) > 0$;

$\psi(x,y) \leq |x|$; $R(x,(y_1,\dots y_n))$ iff $R(\psi(x,y_1,\dots,y_{\sigma(x)}),(y_{\sigma(x)+1},\dots y_n))$

So: we can enumerate the $\sigma(x)$-length segments in poly$(|x|)$ time, and try solving R on $\psi(x,w)$, which has solutions of length $\leq m(x) - \sigma(x) < m(x)$. If we find such a solution $y_{\sigma(x)+1},\dots,y_n$ then we can concatenate w to get a solution for x.

Our satisfiability reduction had $m(\varphi(x_1,\dots,x_n)) = n$, $\sigma(x) = 1$, $\psi(\varphi(x_1,\dots,x_n),b) = \varphi_b(x_1,\dots,x_{n-1})$.

What is a downward self-reduction for clique? Suppose we represent the input as (G,k), and solutions as $(y_1,\dots,y_n)$ s.t. $y_i = 1$ iff i is in a clique of size k.

We can then use $m(G,k) = n$, $\sigma(G,k) = 1$, $\psi((G,k),b)$ is, for $b=1$, $(G_1,k-1)$ where $V(G_1) = \{v \in V(G) : (1,v) \in E(G)\}$, and $E(G_1) = \{(u,v) \in E(G) : u \in V(G_1), v \in V(G_1)\}$. Then a clique of size $k-1$

in G_1 extends to one of size k in G; for $b=0$, (G_0, k) where $V(G_0) = V(G) - \{1\}$, $E(G_0) = E(G) - \{1\} \times V(G)$. Then there was a clique of size k in G that doesn't use 1, that is also in G_0.

Appropriately formulated, this encompasses most of the problems we care about. Now, we sometimes need to count solutions. In general for a relation problem R, the counting version $\#R$ is given x, count how many solutions y there are. Counting also has natural notions of approximation. In particular,

Defn: An algorithm A is an approximate counter for R if $\Pr[(1-\varepsilon)A(x,\varepsilon,\delta)] \le |\{y : (x,y) \in R\}| \le (1+\varepsilon)A(x,\varepsilon,\delta)] \ge 1-\delta$. If A runs in time poly$(|x|, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$ we say A is a fully polynomial randomized approximation scheme (FPRAS)

We'll see that approximate counting for downward self-reducible problems is equivalent to the following problem that is usually easier to tackle, approximate uniform sampling of solutions

Defn: An algorithm A is a fully polynomial almost uniform sampler (FPAUS) for the relation R if for every $S \subseteq Y$, $(1-\varepsilon)\Pr[A(x,\varepsilon) \in S] \le \frac{|\{y \in S : (x,y) \in R\}|}{|\{y \in Y : (x,y) \in R\}|} \le (1+\varepsilon)\Pr[A(x,\varepsilon) \in S]$ and A runs in polynomial time in $|x|$ and $\frac{1}{\varepsilon}$.

Theorem If R is downward self-reducible and has a FPAUS, then R has a FPRAS.

Proof: The downward self-reduction lets us partition the set of solutions based on the value of the first $\sigma(x)$ bits. This gives $\le 2^{\sigma(x)} \le n^{O(1)}$ sets, at least one of which contains at least a $\frac{1}{n^{O(1)}}$ fraction of the solutions. For x and any y, let $R|_{x,y_1 \ldots}$ denote

the set $\{y' : (x,y') \in R, (y'_1, y'_2, \cdots y'_{\sigma(x)}) = (y_1, y_2, \cdots, y_{\sigma(x)})\}$.

Our FPRAS is the following recursive algorithm: for $m = m(x)$,

> Generate $12\left(\frac{m}{\varepsilon}\right)^2 2^{\sigma(x)} \log\frac{2m}{\delta}$ $(1 \pm \frac{\varepsilon}{2m})$-approximate samples.
> Let $y^*_{1:\sigma}$ be the plurality setting of $y_{1:\sigma}$ in these samples, and
> let $\hat{p}$ be the fraction of solutions with value $y^*_{1:\sigma}$.
> Recursively approximately count $\hat{\alpha} = \left(1 \pm (1-\frac{1}{2m})\varepsilon\right) |\{y \in R|_{x, y^*_{1:\sigma}}\}|$
> for the input $\psi(x, y^*_{1:\sigma})$ w.p. $1-(1-\frac{1}{m})\delta$ (if $|y| > \sigma(x)$)
> Return $\frac{1}{\hat{p}}\hat{\alpha}$.

By the Chernoff bound, $\hat{p}$ is a $(1 \pm \frac{\varepsilon}{2m})$ approximation to $\frac{|\{y \in R|_{x,y^*_{1:\sigma}}\}|}{|\{y : (x,y) \in R\}|}$ with probability $1 - 2e^{\frac{1}{3}(\frac{\varepsilon}{2m})^2 2^{-\sigma(x)} \cdot \#\text{samples}} \geq 1 - \frac{\delta}{m}$

by our choice of the # of samples. By a union bound over the correctness of the recursive call and obtaining an accurate est., overall we get that with prob. $1-\delta$, $\frac{1}{\hat{p}}\hat{\alpha} \leq \underbrace{(1+\frac{\varepsilon}{2m})(1+(1-\frac{1}{m})\varepsilon)}_{1+\frac{\varepsilon}{2m}+\varepsilon-\frac{\varepsilon}{m}+\frac{\varepsilon}{2m}(1-\frac{1}{m})\varepsilon \leq 1+\varepsilon} \frac{|\{y:(x,y)\in R\}|}{|R|_{x,y^*_{1:\sigma}}|} |R|_{x,y^*_{1:\sigma}}|$

and similarly, $\frac{1}{\hat{p}}\hat{\alpha} \geq \underbrace{(1-\frac{\varepsilon}{2m})(1-(1-\frac{1}{m})\varepsilon)}_{1-\frac{\varepsilon}{2m}-\varepsilon+\frac{\varepsilon}{m}+\frac{\varepsilon}{2m}(1-\frac{1}{m})\varepsilon > 1-\varepsilon} |\{y:(x,y)\in R\}|$

So: $\frac{1}{\hat{p}}\hat{\alpha}$ is a $(1 \pm \varepsilon)$-approx.

The depth of recursion is $\leq m$ $(\text{poly}(|x|))$ and we run the FPAUS $O\left(\left(\frac{m}{\varepsilon}\right)^2 2^{\sigma(x)} \log\frac{m}{\delta}\right) = \text{poly}(|x|, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$ times at each level. So, the algorithm runs in time $\text{poly}(|x|, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$ overall as needed ▨

This reduction is very useful, because we have techniques for approximate uniform sampling by designing a Markov chain on the set of solutions that converges to a uniform stationary distribution in a small # of steps. These are known as Markov Chain Monte Carlo (MCMC) algorithms. The other direction is conceptually nice, but less practically useful:

Theorem If R is downward self-reducible and has a FPRAS, then R has a FPAUS.

Proof: Recursive algorithm using the FPRAS?

For each setting $y_{1:\sigma}$, use FPRAS on $\psi(x, y_{1:\sigma})$ to get
$\hat{\alpha}(y_{1:\sigma}) = (1 \pm \frac{\varepsilon}{4m}) |R|_{x,y_{1:\sigma}}|$ w.p. $1 - \frac{\delta}{m 2^{\sigma(x)}}$
Put $\hat{\alpha} = \sum_{y_{1:\sigma}} \hat{\alpha}(y_{1:\sigma})$
Choose $y_{1:\sigma}$ w.p. $\frac{\hat{\alpha}(y_{1:\sigma})}{\hat{\alpha}}$
Recursively sample on $\psi(x, y_{1:\sigma})$ to output y.
$(1 \pm (1-\frac{1}{m})\varepsilon)$-approximate w.p. $1-(1-\frac{1}{m})\delta$. (if $\sigma < m$)

We first note that by a union bound on the $2^{\sigma(x)}$ calls to the FPRAS, all yield accurate estimates w.p. $\geq 1 - \frac{\delta}{m}$, and so overall inductively, these counts and the recursive sample are accurate w.p. $1-\delta$.

To see that this is a FPAUS, fix any set S. Let $S|_{y_{1:\sigma}}$ be $\{(y_{\sigma+1}, \ldots, y_m) : (y_1, \ldots, y_\sigma, y_{\sigma+1}, \ldots, y_m) \in S\}$. Inductively, the recursive call returns $(y_{\sigma+1}, \ldots, y_m) \in S|_{y_{1:\sigma}}$ w.p. $(1 \pm (1-\frac{1}{m})\varepsilon) \frac{|S|_{y_{1:\sigma}}|}{|R|_{x,y_{1:\sigma}}|}$, where we sample each $y_{1:\sigma}$ w.p. at least $\frac{1-\frac{\varepsilon}{4m}}{1+\varepsilon/4m} \frac{|R|_{x,y_{1:\sigma}}|}{|\{y:(x,y)\in R\}|} > (1-\frac{\varepsilon}{2m}) \frac{|R|_{x,y_{1:\sigma}}|}{|\{y:(x,y)\in R\}|}$

$((1+\frac{\varepsilon}{4m})(1-\frac{\varepsilon}{2m}) = 1 - \frac{\varepsilon}{4m} - (\frac{\varepsilon}{4m})(\frac{\varepsilon}{2m}) < 1 - \frac{\varepsilon}{4m})$ and similarly, at most $\frac{1+\varepsilon/4m}{1-\varepsilon/4m} \frac{|R|_{x,y_{1:\sigma}}|}{|\{y:(x,y)\in R\}|} < (1 + \frac{\varepsilon}{2m} + \frac{\varepsilon^2}{4m^2})\left((1-\frac{\varepsilon}{4m})(1+\frac{\varepsilon}{2m}+\frac{\varepsilon^2}{4m}) = 1 - \frac{\varepsilon}{4m} + \frac{\varepsilon}{2m} - \underbrace{\frac{\varepsilon^2}{2\cdot 4m} + \frac{\varepsilon^2}{4m}}_{>0} \frac{\varepsilon^3}{4\cdot 4m}\right)$

Hence, overall, the prob. that we get yes is at least
$(1-\frac{\varepsilon}{2m})(1-(1-\frac{1}{m})\varepsilon) = 1 - \varepsilon + \frac{\varepsilon}{m} - \underbrace{\frac{\varepsilon}{2m}}_{>0} + \frac{\varepsilon^2}{2m} - \underbrace{\frac{\varepsilon^2}{2m^2}}_{>0} > 1-\varepsilon$ and at most
$(1 + \frac{\varepsilon}{2m} + \frac{\varepsilon^2}{4m^2})(1+(1-\frac{1}{m})\varepsilon) = 1 + \varepsilon - \underbrace{\frac{\varepsilon}{m} + \frac{\varepsilon}{2m} + \frac{\varepsilon^2}{2m}}_{<0} - \underbrace{\frac{\varepsilon^2}{2m^2} + \frac{\varepsilon^2}{4m^2} + \frac{\varepsilon^3}{4m^2}}_{<0} - \underbrace{\frac{\varepsilon^3}{4m^3}}_{<0} < 1+\varepsilon$
times $\frac{|S|}{|\{y:(x,y)\in R\}|}$ as needed.