CS880: Approximations Algorithms

Scribe: Matt Darnall **Lecturer:** Shuchi Chawla
Topic: Apporximate Counting **Date:** 4/12/07

24.1 Definitions

We will attempt to approximate a function that counts something. Typically, we are interested in finding the number of solutions to an NP-Complete problem, which is of course harder than solving the NP-Complete problem itself. So many such problems are # P-complete. An example of this is the number of ways to satisfy a statement in disjunctive normal form. A good algorithm for approximating a function will be an FPRAS or Fully Polynomial Randomized Approximation Scheme.

An algorithm A is a FPRAS for a function f if given an instance X of the problem, for any ϵ,

$$Pr\left[\frac{|A(X)-f(X)|}{f(X)} < \epsilon\right] < 1/4$$

and the algorithm runs in time polynomial in the length of X and $1/\epsilon$. The algorithm is just a PRAS if it runs in time polynomial in the length of X, but not in $1/\epsilon$.

What we are looking for is an algorithm A such that:

$$Pr\left[A(X) \in ((1-\epsilon)f(X), (1+\epsilon)f(X))\right] < 1-\delta$$

where the runtime is polynomial in the size of X, $1/\epsilon$, and $log(1/\delta)$. By iterating, an FPRAS will give this to us.

24.2 Disjuntive Normal Form Counting

The problem we shall look at is counting the number of truth assignments to variables x_i such that the statement D is satisfied. D is forced to look like $D_1 \vee \cdots \vee D_m$, where each D_i is a conjunction of variables $x_{i_1} \wedge \cdots \wedge x_{i_k}$.

Let U be the set of all assigments, S be the set of satisfying assignments. We shall approximate $|S|$ by randomizing selecting elements of U and testing for membership in S. Then, if we test t samples from U and t' are from S, we know that:

$$\frac{|S|}{|U|} \approx \frac{t'}{t}$$

so our estimate is $|S| = t'|U|/t$. If $p = |S|/|U|$ is our probability of pulling a member of S, we get by Chernoff bounds that:

$$Pr\left[|t'-pt| > \epsilon pt\right] < e^{\frac{\epsilon^2 pt}{3}}$$

Thus, taking $t = \log(1-\delta)/(\epsilon^2 p)$ will get us within $1 \pm \epsilon$ of $|S|$ with probability greater than $1-\delta$. The only thing that can make this not polynomial time is if p is small. This means that $|S|$ is small relative to $|U|$.

If we use Chebyshev's inequality instead of Chernoff, we get that:

$$Pr\left[|t' - pt| > \epsilon pt\right] < \frac{Var(t')}{\epsilon^2 p^2 t^2}$$

Since $Var(t') = p(1-p)t$, we need to take $t = (1-p)/(\epsilon^2 p\delta)$. In order to reduce the dependence on δ, we use the "'median of means"' method. Now, if we let $\delta = 1/4$, and we run the experiment with $t = 4(1-p)/(\epsilon^2 p\ 2\Delta + 1$ times, we expect about $(2\Delta + 1)/4$ of our trials to fall out of the $(1 \pm \epsilon)$ range of the actual solution. Specifically, the probability that the median of the $2\Delta + 1$ trials with $t = 4(1-p)/(\epsilon^2 p$ falls outside the $(1 \pm \epsilon)$ range of the actual solution is less than the probability that $\Delta + 1$ of the trials falls out of the $(1 \pm \epsilon)$ range of the actual solution. This probability, by Chernoff, is less than $(3/4)^s$. Thus, by running t trials $O(log(1/\delta))$ times, and taking the median of these answers, we are within $(1 \pm \epsilon)$ of the acutal solution with probability $1-\delta$.

24.3 Reducing U

It is evident when analyzing the value of t that making the ratio $p = |S|/|U|$ small is crucial to limiting the runtime. We shall look at a method for reducing the universe U for our example of solutions to a DNF statement. Let the statement D be made up of m clauses D_i that are conjuntions of the literals or their negation. Let the solutions of D_i be S_i. Notice that S is just the union of the S_i and each S_i has size 2^{n-k_i}, where there are k_i literals in D_i and n literals in total. Although we can easily estimate $|S_i|$, it is hard to get an estimate of their union because of repitition.

Our new universe, U' shall be the set of pairs (i, a_i), $i \leq m$ where a_i is an element of S_i. Notice that sampling from U' at random is easy, we just pick a random i and then pick a random assignment to the variables not in D_i. In order for this to be uniform, we pick i with probability proportional to $|S_i|$, we we can do since cacluting $|S_i|$ is easy (just 2 to the power of the number of literals in S_i). The variables in D_i are fixed. We need a way to imbed the set S inside our new universe U' and a quick way to test membership in S. To do this, let every assignment, s, of the variables that satisfies D be associated with the first D_i that is satisfies. In other words, our embedded S' is the set of (i, s), where $s \in S$ and $i \leq m$ is the first D_i that s satisfies. Since it satisfies at least one of the D_i, we know that this assignment is counted exactly once in the set U', so S' is of the same size as S. Importantly, since there are only m clauses, and each element in S' is associated with a clause, $|S'|/|U'|$ is greater than $1/m$. Thus, we have made p larger than the inverse of the size of the instance, and our number of trials t can be polynomial in the size of the instance.

24.4 Counting = Sampling

In this section, we will give an overview of the proof that being able to approximately count the elements in S is equivalent to being able to uniformly sample from S. Here S is the solution set of a finite problem. The problem must be self reducible.

We define a problem Π to be *self reducible* if, given an instance P of size n:

1. The solutions that satisfy P can be written as binary strings in $poly(n)$. S_i is the set of solutions with the first bit of the string i.
2. There is a problem $P' \in \Pi$ of length less than n such that there is a bijection of the solutions in P' to the solutions S_0 (and so also S_1.

Theorem 24.4.1 *If* Π *is self reducible, then there is a near uniform way to sample solutions iff there is an FPRAS for* Π.

Near uniform way to sample solutions means the probability of selecting a particular solution is in between $(1-\epsilon)/t$ and $(1+\epsilon)/t$, where there are t solutions total.

If you can sample near uniformly, then we can use the fact that:

$$|S| = \frac{|S|}{|S_0|}\frac{|S|_0}{|S_{00}|}\cdots$$

to estimate $|S|$ using our uniform sampling and self reducibility to estimate the ratios on the right hand side. We will have a good estimate on $|S|$ if we had a good way of sampling.

If we have an FPRAS for Π, then we estimate the size of S_0 and S_1 using our FPRAS. We then select the first bit of our random solution to be 0 with probability $\frac{|S_0|}{|S_1|}$. We then recursively pick the next bits of our random solution in the same way. The fact that our FPRAS is able to approximate the number of solutions with a certain starting sequence of bits well means that we get a close to uniformly picked solution.

24.5 Further Reading

A great set of lecture slides on this material can be found at:

http://www3.math.tu-berlin.de/ipco05/Pages/Download/IPCO05_Dyer_LectureNotes.pdf