

CS 155 HW 9

Gabe Bankman-Fried

April 28, 2016

Problem 1, exercise 10.5

Part a

Consider the following slightly different algorithm for estimating $|S|$:

For every element which appears in multiple sets S_i, S_j, \dots , keep one copy of that element the same and label each of the other copies of that element differently. Now we have produced disjoint sets S'_1, S'_2, \dots . Call their union S' . Note that $S \subset S'$, and note that $|S'| = |S'_1| + |S'_2| + \dots$ since the sets are disjoint, so since we know $|S'|$, we can use Monte Carlo sampling as described in the lecture slides. This obtains a bound of:

$$N > \frac{3}{\rho\epsilon^2} \log\left(\frac{2}{\delta}\right).$$

Now I claim that this bound also applies in our situation. To see that, note that in the algorithm above, we are sampling an item x_j uniformly and only increasing our count by one if it's in a particular one of $c(x_j)$ sets. So in the Monte Carlo algorithm above, we increase the count by 1 with probability $\frac{1}{c(x_j)}$.

In our algorithm, we increase the count by $\frac{1}{c(x_j)}$ with probability 1, which is just a lower variance way of approximating the same quantity! In other words, the algorithm given in the homework is strictly noisier than the algorithm given above, so the algorithm above would require more samples than our algorithm. Therefore the above bound also holds for our algorithm, meaning to get an (ϵ, δ) -approximation, we require:

$$N > \frac{3}{\rho\epsilon^2} \log\left(\frac{2}{\delta}\right).$$

Part b

We can count solutions to a DNF formula the same way we counted members of S in part a. For each clause in our DNF, set all of its variables to True and then let S_i be the set of all truth assignments for the rest of the variables in our DNF. Then we are interested in the size of the union of each S_i , call that S . Note that we have access to the same tools as in part a. We know the size of each S_i , since it is just 2^{V_i} , where V_i is the number of variables in every clause other than the i^{th} one. And we know $c(x_i)$, the number of sets S_i in which a clause appears. To find this, we can just see how many of the clauses that x_i satisfies, and this will necessarily be the number of sets S_j such that $x_i \in S_j$.

So we can use our algorithm from part a. In other words, we sample a truth assignment t times uniformly from all of our S_i s, and then estimate $|S|$ by:

$$\left(\frac{1}{t} \sum_{j=1}^t \frac{1}{c(x_i)}\right) \left(\sum_{i=1}^m |S_i|\right).$$

The bound in part a can tell us how large t should be if we want to use this algorithm to get an (ϵ, δ) -approximation of $|S|$.

Problem 2, exercise 10.8

Define $N(x)$ to be $\{2\}$ if $x = 1$ and $\{x \pm 1\}$ otherwise. Define a Markov chain by the following transition probabilities from state x to y :

$$P_{x,y} = \begin{cases} \frac{\min(1, \frac{x^2}{y^2})}{2} & y \in N(x) \\ 1 - \sum_{z \in N(x)} \frac{\min(1, \frac{x^2}{z^2})}{2} & y = x \\ 0 & otherwise \end{cases}$$

Because the maximum number of neighbors of any state is 2, setting $M = 2$ and $\pi_x = \frac{1}{S^2}$ (where S is as described in the problem) lets us apply Lemma 10.8 in the book. So as long as the Markov chain defined by the transition probability stated above is aperiodic and irreducible, then it must have stationary state distribution $\pi_x = \frac{1}{S^2}$. A Markov chain is aperiodic if there exists some state which can transition to itself. 1 is one such state, since $P_{1,2} = \min(1, \frac{1^2}{2^2}) = 1/4$, so $P_{1,1} = 1 - 1/4 = 3/4 > 0$. Therefore, this Markov chain is aperiodic. A Markov chain is irreducible if there exists a path with nonzero probability from every state i to every state j . Well such a path definitely exists, since this Markov chain just does a one-step walk along the positive integers, so we can always move up or down by 1, meaning to get from i to j , if $i > j$, we can take the path $i, i - 1, \dots, j$. If $i < j$, we can take the path $i, i + 1, \dots, j$, and if $i = j$ we can take the path $i, i + 1, i = j$. So this Markov chain is aperiodic and irreducible, meaning by Lemma 10.8, it has stationary distribution:

$$\pi_x = \frac{1}{S^2}.$$

Problem 3, exercise 10.10

Here we can use the same procedure as in problem 2. Given a graph G , let our state space Ω be the set of all colorings of the vertices of G . Now for all $x \in \Omega$, define $N(x)$ to be the set of all colorings which differ from x by exactly vertex. Note that $N(x) = \Delta$, so we can

pick $M = \Delta + 1$ when using Lemma 10.8. Let $I(x)$ be the number of improper edges in x . If we want to construct a Markov chain with stationary distribution $\pi_x \propto \lambda^{I(x)}$, we can do it as follows:

Let $1/Z$ be the normalizing constant such that $\sum_x \lambda^{I(x)} = Z$. Then Lemma 10.8 tells us that if the distribution given below is aperiodic and irreducible, it has stationary distribution $\pi_x = \lambda^{I(x)}/Z$.

$$P_{x,y} = \begin{cases} \frac{\min(1, \frac{\lambda^{I(y)}}{\lambda^{I(x)}})}{\Delta+1} & y \in N(x) \\ 1 - \sum_{z \in N(x)} \frac{\min(1, \frac{\lambda^{I(z)}}{\lambda^{I(x)}})}{\Delta+1} & y = x \\ 0 & \text{otherwise} \end{cases}$$

This Markov chain is aperiodic because we picked $M = \Delta + 1 > \Delta$, so every coloring x has some probability of staying the same. This Markov chain is also irreducible since we can obviously get from every coloring to every other coloring just by changing the color of one vertex at a time. Therefore, Lemma 10.8 tells us that the above Markov chain has the desired stationary distribution.

Note: our normalization constant Z canceled out in the application of Lemma 10.8, so our answer doesn't depend on it. Which is good, because it would be incredibly hard (some might even call it NP-hard) to calculate.