CSE 632 (Fall 2019): Analysis of Algorithms II : Randomized Algorithms

Lecture 19 (10/30/2019): Randomized 3-SAT

Lecturer: Shi Li *Scriber: Yifu Yin*

1 3-SAT

In the last lecture, we talked about the randomized algorithm for 2-SAT using the analysis for the random walk on a line. In this lecture, we talk about randomized algorithm for 3-SAT algorithms. The 3-SAT problem is the same as 2-SAT, except that each clause contains 3 literals. Unlike 2-SAT, which is a problem in P, the 3-SAT problem is NP-complete and thus it is unlikely that it can be solved in polynomial time. Indeed, there is a conjecture stating that it can not be solved in sub-exponential time:

Hypothesis 1 (Exponential Time Hypothesis (ETH))**.** *3-SAT can not be solved in time $2^{o(n)}poly(n,m)$ time algorithm, where n and m are the number of variables and clauses in the instance respectively.*

A trivial algorithm to solve 3-SAT, is to enumerate all the 2^n assignment of variables, then check if formula is satisfied. Runtime for this trivial algorithm: $O(2^n \times mm)O(2^n\text{poly}(n,m))$. In the lecture we shall give a randomized algorithm that runs in time $O((4/3)^n\text{poly}(n,m))$, suggesting that we can improve the constant in base of the exponential function. However, the ETH states that this is the only type of improvements we can get: It does not contradict ETH that there exists a $1.001^n\text{poly}(n,m)$-time algorithm, but an algorithm with running time $2^{n^{0.99}}\text{poly}(n,m)$ does.

2 Randomized Algorithm for 3-SAT

The randomized algorithm works as follows:

```
for up to $\frac{3e^2\sqrt{n}}{\sqrt{3\pi}}\left(\frac{4}{3}\right)^n$ iterations do
    randomly generate an assignment of $\{0,1\}$ values to the $n$ varaibles
    for up to $3n$ time steps, terminating if all clauses are satisfied do
        pick an arbitrary clause $c$ that is not satisfied by the assignment $x$
        randomly choose 1 of 3 variables in the clause and flip its $x$ value
    if all clauses are satisfied then return true
return false
```

2.1 Correctness Analysis and success probability

If the instance is unsatisfiable, then our algorithm always returns false since it returns true only if we found an satisfying assignment x. On the other hand, if the instance is satsifiable, the algorithm may return false as it may fail to find a satisfying assignment. So, the algorithm may return a false negative, but not a false positive. Our goal is then to show that the probability that the algorithm returns false is small, assuming that the instance is satisfiable.

To the end of the analysis, we fix a satisfying assignment x^* for the instance. We fix an iteration of the outer-loop and analyze its success probability. As in the analysis of 2-SAT, we let Y_t be the number of variables where $x_i = x^*_i$ after t time steps of the inner loop.

First, Y_0 follows the binomial distribution with parameters n and $1/2$. That is, the probability that $Y_0 = i$ is $\binom{n}{i}/2^n$ for every $i \in [0,n]$. Let us $t \geq 0$ and $Y_0, Y_1, \cdots, Y_t$. Then, Y_{t+1} is either $Y_t - 1$ or $Y_t + 1$. Moreover, we have

$$Y_{t+1} = \begin{cases} Y_t - 1 \text{ with probability at most } 2/3 \\ Y_t + 1 \text{ with probability at least } 1/3 \end{cases}$$

This holds since every time we choose an unsatisfied clause c and flip one of the 3 variables in c uniformly at random. Since x^* is a satisfying assingment, c is satisfied by x^*. Thus, at least one of the three variables have different x and x^* values. Also, the iteration will terminate successfully if for some t we have $Y_t = n$ (the iteration may terminate earlier since there might be satisfying assignments other than x^*).

Thus, the success probability of an iteration is at least the probability that the following procedure returns "success":

```
Choose $Y_0$ randomly from the binomial distribution with parameters $n$ and $1/2$
for $t \leftarrow 1$ to $3n$ do
    with probability $1/3$ let $Y_t \leftarrow Y_{t-1} + 1$ (we choose "right")
    and with the remaining probability $2/3$, let $Y_t \leftarrow Y_{t-1} - 1$ (we choose "left")
return "success" if $Y_t = n$ for some $t \in [0, 3n]$ and "failure" otherwise
```

Let us condition on the even that $Y_0 = n - j$ for some integer $j \in [0, n]$. Then the probability that the above procedure returns success is at least the probability that in j out of the first $3j$ steps, we choose "right" and in the remaining $2j$ steps we choose "left". The probability is

$$\binom{3j}{j}\left(\frac{1}{3}\right)^{2j}\left(\frac{2}{3}\right)^{j}.$$

To obtain an acurate estimate of the term, we approximate the binormial coefficient using Stirling's approximaton for factorials:

$$\sqrt{2\pi m}\left(\frac{m}{e}\right)^m \le m! \le e\sqrt{m}\left(\frac{m}{e}\right)^m$$

Then we substitute all factorials using the inequalities, we get

$$\begin{aligned}\binom{3j}{j}\left(\frac{1}{3}\right)^{2j}\left(\frac{2}{3}\right)^{j} = \frac{(3j)!}{j!(2j)!}\left(\frac{2}{27}\right)^{j} &\ge \frac{\sqrt{2\pi\cdot 3j}(\frac{3j}{e})^{3j}}{e\sqrt{2j}(\frac{2j}{e})^{2j}\cdot e\sqrt{j}(\frac{j}{e})^{j}}\left(\frac{2}{27}\right)^{j}\\ &= \frac{\alpha}{\sqrt{j}}\frac{3^{3j}}{2^{2j}}\left(\frac{2}{27}\right)^{j} \ge \frac{\alpha}{\sqrt{n}}\left(\frac{1}{2}\right)^{j},\end{aligned}$$

whereas $\alpha = \frac{\sqrt{6\pi}}{\sqrt{2}e^2} = \frac{\sqrt{3\pi}}{e^2}$ is an absolute constant.

Then we consider all possible values that Y_0 can take, the probability that an iteration of the outer loop is successful with probability at least

$$\frac{\alpha}{\sqrt{n}}\sum_{j=0}^{n}\frac{\binom{n}{j}}{2^n}\left(\frac{1}{2}\right)^{j} = \frac{\alpha}{\sqrt{n}}\frac{1}{2^n}\left(1+\frac{1}{2}\right)^{n} = \frac{\alpha}{\sqrt{n}}\left(\frac{3}{4}\right)^{n}.$$

One iteration of the outer loop fails with probability at most $1 - \frac{\alpha}{\sqrt{n}}(\frac{3}{4})^n \le e^{-\frac{\alpha}{\sqrt{n}}(\frac{3}{4})^n}$. The $m := \frac{3\sqrt{n}}{\alpha}\left(\frac{4}{3}\right)^n = \frac{3e^2\sqrt{n}}{\sqrt{3\pi}}\left(\frac{4}{3}\right)^n$ iterations all fail with probability at most $e^{-3} < 0.1$. Thus the algorithm succeeds with probability at least 0.9. By increasing the number of iterations of the outer loop by a factor of $\log n$, the success probability can be increased to $1 - 1/n^2$.

Remark on Two Different Markov Chains on a Line Consider the following Markov chain on a line. We start from $Y_0 = 0$. For each $t = 1, 2, 3, \cdots$, we let $Y_t = Y_{t-1} - 1$ with probability p and $Y_t = Y_{t-1} + 1$ with probability $1 - p$. There are two terminating thresholds $-A$ and B where $A, B > 0$ are integers: When $Y_t = -A$ for some t the procedure terminates and we say the player A wins the game; when $Y_t = B$ for some t the procedure also terminates and we say the player B wins the game.

We consider two different settings for the Markov Chain:

1. $p = 1/2$, $A = n$ and $B = 2n$ where n is a parameter that tends to ∞. This game is in favor of A in the sense that the terminal A is closer to the starting point than that for B. In this case we can show that the probability that A wins the game is exactly $2/3$, which is independent on n.

2. $p = 2/3$, $A = B = n$ where n is a parameter that tends to ∞. This game is again in favor of A since the probabilities of going left and right are biased towards A. In this case, however, the probability that A wins the game is much larger than that of B winning the game. Using Chernoff bounds, we can show that the probability that B wins the game is exponentially small in n.