COMPSCI 630: Randomized Algorithms	January 22, 2018
Lecture 2	
Lecturer: Rong Ge	*Scribe: Kevin Sun*

1 Overview

Last lecture, we began studying of randomized algorithms, motivated their utility, and saw two randomized algorithms for the minimum cut problem. In this lecture, we will go over some basic definitions, Markov's inequality, and the difference between Las Vegas and Monte Carlo algorithms. We will also see a randomized algorithm for the maximum satisfiability problem and analyze its performance.

2 Random Variables and Expectation

In this section, we review some basic definitions and properties involving random variables and expectation. Throughout, we let X be a real-valued random variable and Ω denote the set of possible values X can take.

Definition 1. *The* expectation *of X, denoted by $\mathbb{E}[X]$, is given by*

$$\mathbb{E}[X] = \sum_{i \in \Omega} i \cdot \Pr(X = i)$$

if Ω is a discrete set, and

$$\mathbb{E}[X] = \int_x x \cdot p(x)\, dx,$$

where p is the density function of X, if Ω is continuous.

In this class, we'll mostly be working with discrete random variables. The following theorem is a fundamental property of random variables that is frequently used in the analysis of randomized algorithms.

Theorem 1 (Linearity of Expectation)**.** *If X and Y are discrete random variables with finite expectations, then*

$$\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y].$$

Proof. To prove the statement, we simply rearrange the sum given by the definition of expectation and collect terms appropriately. Let Ω_X and Ω_Y denote the sets of possible values for X and Y, respectively. Note that

$$\begin{aligned}
\mathbb{E}[X+Y] &= \sum_{\substack{i \in \Omega_X \\ j \in \Omega_Y}} (i+j) \Pr(X = i, Y = j) \\
&= \sum_{i \in \Omega_X} i \sum_{j \in \Omega_Y} \Pr(X = i, Y = j) + \sum_{j \in \Omega_Y} j \sum_{i \in \Omega_X} \Pr(X = i, Y = j) \\
&= \sum_{i \in \Omega_X} i \cdot \Pr(X = i) + \sum_{j \in \Omega_Y} j \cdot \Pr(Y = j) \\
&= \mathbb{E}[X] + \mathbb{E}[Y].
\end{aligned}$$

Note that third line holds due to the law of total probability. □

By induction, we can extend Theorem 1 to any finite collection of discrete random variables. Also, notice that the proof of Theorem 1 requires no assumptions on X or Y, so in particular, linearity of expectation holds regardless of the correlation among the variables under consideration.

We now give an application of linearity of expectation by giving an algorithm for the maximum satisfiability problem, which we abbreviate as MAX-3SAT. The input consists of n Boolean variables $x_1, \ldots, x_n$ and m clauses $C_1, \ldots, C_m$. We assume each clause is a disjunction ("or") of 3 literals (a variable x_i or its negation $\overline{x_i}$). Furthermore, we assume these literals correspond to distinct variables. For example, $x_1 \vee x_3 \vee \overline{x_4}$ is a valid clause, but $x_2 \vee x_2 \vee x_5$ is not. We say a clause is satisfied if one of its unnegated variables is set to true or one of its negated variables is set to false. Our objective is to find an assignment of true/false to the x_i that maximizes the number of satisfied clauses. Let OPT denote the maximum number of clauses that can be simultaneously satisfied in the MAX-3SAT instance.

Theorem 2. *For the MAX-3SAT problem, there exists a randomized algorithm whose expected number of satisfied clauses is at least* $7/8 \cdot OPT$.

Proof. The algorithm is simple: set each variable x_i to be true with probability $1/2$ independently. Now let Y_j be a random variable such that Y_j is 1 if the clause C_j is satisfied and 0 otherwise. Then we have

$$\Pr(Y_j = 0) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8}$$

because all 3 variables involved in C_j must fail simultaneously for C_j to not be satisfied. Thus, we also have

$$\mathbb{E}[Y_j] = \Pr(Y_j = 1) = 1 - \frac{1}{8} = \frac{7}{8}.$$

Now let N denote the number of clauses satisfied by our algorithm, so

$$N = \sum_{j=1}^{m} Y_j.$$

Taking the expectation of both sides and applying linearity of expectation yields the following:

$$\mathbb{E}[N] = \mathbb{E}\left[\sum_{j=1}^{m} Y_j\right] = \sum_{j=1}^{m} \mathbb{E}[Y_j] = \sum_{j=1}^{m} \frac{7}{8} = \frac{7m}{8} \geq \frac{7}{8}\text{OPT},$$

as desired. Note that the Y_j are not necessarily independent, but as mentioned above, linearity of expectation still holds. □

It is remarkable that no better approximation algorithm exists for MAX-3SAT unless P = NP.

3 Markov's Inequality

In the proof of Theorem 2, we introduced the concept of an indicator random variable, that is, a random variable that takes value 1 if some event occurs, and 0 otherwise. This enables us to equate the probability of an event E occurring with the expectation of its indicator random variable. By doing this, we can exploit linearity of expectation and possibly avoid more complicated expressions involving probabilities.

So it is natural to ask if we can go the other way, that is, if we can use the expectation of a random variable to bound the probability of some event occurring. One of the fundamental tools here is known as Markov's inequality.

Theorem 3 (Markov's Inequality)**.** *Let X be a random variable that is nonnegative with probability* 1*. Then for all $t > 0$,*

$$\Pr(X \geq t) \leq \frac{\mathbb{E}[X]}{t}.$$

Proof. Let Y be the indicator random variable for the event $X \geq t$. Since $X \geq 0$ with probability 1, we have $Y \leq X/t$, and taking expectations yields the result:

$$\Pr(X \geq t) = \mathbb{E}[Y] \leq \mathbb{E}\left[\frac{X}{t}\right] = \frac{\mathbb{E}[X]}{t}.$$

□

We now apply Markov's inequality to MAX-3SAT. Recall that Theorem 2 gives us an algorithm whose expected number of satisfied clauses is at least $7/8 \cdot \mathrm{OPT}$. However, this does not give us the probability of actually obtaining a decent assignment; the following theorem addresses this concern.

Theorem 4. *There exists a randomized algorithm for MAX-3SAT that runs in $O((m+n)/\varepsilon)$ time and satisfies $\frac{7-\varepsilon}{8}m$ clauses with probability at least $2/3$, where m is the total number of clauses and ε is any real number in $(0, 1/2)$.*

Proof. Our algorithm is the same as before: independently assign each x_i to be true with probability $1/2$. Let N again denote the number of satisfied clauses. Notice that we want to lower bound the value of

$$\Pr\left(N \geq \frac{7-\varepsilon}{8}m\right),$$

so Markov's inequality does not immediately apply. We can remedy this by considering the number of clauses not satisfied by the algorithm, which we denote by Z. Note that $N + Z = m$, so

$$\Pr\left(N \geq \frac{7-\varepsilon}{8}m\right) = \Pr\left(Z \leq \frac{1+\varepsilon}{8}m\right).$$

Letting Y_j be the indicator random variable for clause C_j being satisfied, we have

$$\mathbb{E}[Z] = \sum_{i=1}^{m} \mathbb{E}[1 - Y_j] = \frac{m}{8}.$$

By Markov's inequality, we have

$$\Pr\left(Z > \frac{1+\varepsilon}{8}m\right) \leq \frac{\mathbb{E}[Z]}{\frac{1+\varepsilon}{8}m} = \frac{1}{1+\varepsilon} < 1 - \frac{\varepsilon}{2},$$

where the finality inequality holds since we've assumed ε is in $(0, 1/2)$. Therefore,

$$\Pr\left(Z \leq \frac{1+\varepsilon}{8}m\right) > \frac{\varepsilon}{2}.$$

One iteration of the algorithm involves assigning truth values to n variables and checking which of the m clauses are satisfied. Thus, running the algorithm $O(1/\varepsilon)$ times and returning the best assignment, for a total runtime of $O((m+n)/\varepsilon)$, proves the theorem. □

4 Las Vegas and Monte Carlo

We end the lecture by discussing the relationship between two different types of randomized algorithms.

Definition 2. *An algorithm is a* Las Vegas algorithm *with runtime $T(n)$ if it always gives a correct answer and its expected runtime is at most $T(n)$.*

Definition 3. *An algorithm is a* Monte Carlo algorithm *with runtime $T(n)$ if it always has runtime at most $T(n)$ and returns a correct answer with probability at least $2/3$.*

Note that in Lecture 1, we used the Monte Carlo definition for randomized algorithms. This is because we can easily convert any Las Vegas algorithm into a Monte Carlo algorithm, as we shall see in the following theorem.

Theorem 5. *If A is a Las Vegas algorithm with runtime $T(n)$, then there exists a Monte Carlo algorithm A' for the same problem with runtime $3T(n)$.*

Proof. Our proposed algorithm A' is as follows: run A until either it outputs the correct answer, or more than $3T(n)$ time steps have elapsed. The probability that the runtime of A exceeds $3T(n)$ is, by Markov's inequality, at most $1/3$. Thus, A' has runtime $3T(n)$ and returns a correct answer with probability at least $2/3$, so it is Monte Carlo, as desired. □

5 Summary

In this lecture, we covered linearity of expectation, Markov's inequality, and Las Vegas algorithms. We also saw a randomized algorithm for maximum satisfiability and analyzed its performance two different ways.