Lecture 7:
Introduction to Randomized Algorithms

Motwani and Raghavan *Randomized Algorithms* p. 103

Williamson, *Lecture Notes on Approximation Algorithms* 1999

IBM Research Report RC21409, pp 46-47

Can be found at http://www.almaden.ibm.com/cs/people/dpw/

Quick Review of Probability Theory

A *random variable* is a real number that is the outcome of a random event. For example, X can be the number of spots showing after throwing a die.

The *expectation* of X is

$$E[X] = \sum_i i \cdot Pr(X = i) = \int \alpha f_X(\alpha) d\alpha;$$

The first equation is used if X is discrete (the sum is over all possible values of X); the second if X is continuous ($f_X(\alpha)$ is the *density function of X*).

Some basic facts.

- If X and Y are *any* two random variables then $E[X + Y] = E[X] + E[Y]$.
- If c is any number then $E[cX] = cE[X]$.
- If X and Y are *independent* then $E[XY] = E[X] \cdot E[Y]$.

Example 1: Throw two dice. Let X and Y be the respective number of spots showing on each of them.

$$E[X] = E[Y] = \sum_{i=1}^{6} iPr(X = i) = \sum_{i=1}^{6} \frac{i}{6} = \frac{7}{2}.$$

Therefore the expected sum of the two dice's spots is

$$E[X + Y] = E[X] + E[Y] = \frac{7}{2} + \frac{7}{2} = 7.$$

Since X and Y are independent

$$E[XY] = E[X] \cdot E[Y] = \frac{7}{2} \cdot \frac{7}{2} = \frac{49}{4}$$

Example 2: Now define

$$A = \begin{cases} 1 & X \text{ is even} \\ 0 & X \text{ is odd} \end{cases}, \qquad B = \begin{cases} 0 & X \text{ is even} \\ 1 & X \text{ is odd} \end{cases}$$

Calculation shows that $E[A] = E[B] = \frac{1}{2}$,
$A + B = 1$ and $AB = 0$.

Notice

$$E[A + B] = 1 = \frac{1}{2} + \frac{1}{2} = E[A] + E[B]$$

but

$$E[AB] = 0 \neq \frac{1}{4} = \frac{1}{2} \cdot \frac{1}{2} = E[A] \cdot E[B].$$

This is because A and B *are not* independent.

Indicator Random Variables

Let W be some of *event*. The *indicator random variable* of W is the function

$$I_W = \begin{cases} 1 & \text{if } W \text{ happens} \\ 0 & \text{if } W \text{ does not happen} \end{cases}$$

For example. Suppose we throw a die and let X be the number of spots that show. W could be the event "X is even". Then

$$I_W = \begin{cases} 1 & X \text{ is even} \\ 0 & X \text{ is odd} \end{cases}$$

and I_W is the random variable A defined on the previous page. The important fact about indicator random variables is

$$E[I_W] = Pr(W \text{ happens}).$$

Example: Suppose n people all having the same style coat go to the same party. When they leave the party they take the first coat they see that looks like their coat. What is the expected number of people that get their *own* coat back?

Let W_i be the event that person i gets their own coat back. Since every person is equally likely to get person i's coat $Pr(W_i) = \frac{1}{n}$.

Now let

$$X = \text{ No. of people who get their own coat} = \sum_{i=1}^{n} I_{W_i}.$$

Then

$$\begin{aligned} E[X] &= E\left[\sum_{i=1}^{n} I_{W_i}\right] \\ &= \sum_{i=1}^{n} E\left[I_{W_i}\right] \\ &= \sum_{i=1}^{n} Pr(W_i) = \sum_{i=1}^{n} \frac{1}{n} = 1. \end{aligned}$$

So the expected number of people who get their own coat back is 1.

Example:

Let $G = (V, E)$ be a graph with $V = \{1, \ldots, n\}$. Now pick a subset $S \subseteq V$ *at random* using the following procedure.

> $S = \emptyset$.
> For $i = 1$ to n do
>   Flip a fair coin. If "heads", set $S = S \cup \{i\}$.

What is the expected number of edges in the cut
$\delta(S) = (S, V - S) = \{(u, v) \,:\, u \in S,\, v \in V - S\}$?

Define

$$I_{i,j} = \begin{cases} 1 & \text{if } i \in S \text{ and } j \notin S \\ 0 & \text{otherwise} \end{cases}.$$

Then

$$X = \sum_{1 \le i,j \le n,\, i \ne j} I_{i,j} = \delta(S).$$

Note that

$$E\left[I_{i,j}\right] = \Pr(i \in S \text{ and } j \notin S) = \frac{1}{4}$$

so

$$E[\delta(S)] = \sum_{1 \le i,j \le n,\, i \ne j} E\left[I_{i,j}\right] = \sum_{1 \le i,j \le n,\, i \ne j} \frac{1}{4} = \frac{n(n-1)}{4}.$$

A Randomized Approximation Algorithm for Max Cut

Recall the max-cut problem. We are given a weighted graph $G = (V, E)$ and want to find a *cut* $S \subseteq V$ with maximum value $\delta(S)$. The value of a cut was defined to be

$$\delta(S) = \sum_{(u,v)\,:\, u \in S, v \in V - S} w(u, v).$$

The value of an optimal cut is defined to be OPT.

Choose a random cut as defined previously:

> $S = \emptyset$.
> For $i = 1$ to n do
>   Flip a fair coin. If "heads", set $S = S \cup \{i\}$.

Lemma: $E[\delta(S)] \ge \frac{OPT}{2}$.
Proof: Let $I_{i,j}$ be defined on the previous page. Then

$$\delta(S) = \sum_{1 \le i,j \le n,\, i \ne j} w(i, j) I_{i,j}.$$

Therefore

$$\begin{aligned} E[\delta(S)] &= E\left[\sum_{1 \le i,j \le n,\, i \ne j} w(i, j) I_{i,j}\right] \\ &= \sum_{1 \le i,j \le n,\, i \ne j} w(i, j) E\left[I_{i,j}\right] \\ &= \frac{1}{4} \sum_{1 \le i,j \le n,\, i \ne j} w(i, j) \\ &= \frac{1}{2} \sum_{1 \le i < j \le n} w(i, j) \ge \frac{1}{2} OPT. \end{aligned}$$

The MAX SAT problem

Let $x_1, x_2, \ldots, x_n$ be BOOLEAN variables. These variables are set to be either TRUE (T) or FALSE (F). A variable x_i is T if and only if its negation $\bar{x}_i$ is F and vice-versa.

A *clause* is the conjunction of random variables and their negations, e.g., $x_1 \vee \bar{x}_3 \vee x_4$.

Given a *truth assignment* for the $x_1, x_2, \ldots, x_n$ a clause is *satisfied* if at least one of its elements is T. For example, $x_1 \vee \bar{x}_3 \vee x_4$ is satisfied if $x_1 = T$, $x_3 = F$ or $x_4 = T$.

Given n boolean variables, m clauses C_i, $i = 1, \ldots, m$ over those variables and a weight $w_i \ge 0$ for each clause the MAX SAT problem is to find a truth assignment for the variables that maximizes the total weight of the clauses satisfied. This problem is NP hard.

Random MAX SAT
For $i = 1$ to n do
Flip a fair coin.
If "heads", set $x_i = T$.
else set $x_i = F$.

Lemma: Let OPT be the weight of the optimal assignment and W the weight of the random assignment. Then

$$E[W] \geq \frac{OPT}{2}.$$

Proof: Set

$$I_j = \begin{cases} 1 & \text{if } C_j \text{ is satisfied} \\ 0 & \text{otherwise} \end{cases}.$$

Let l_j be the number of variables in C_j. Then

$$\begin{aligned} E[W] &= E\left[\sum_j w_j I_j\right] \\ &= \sum_j w_j E[I_j] \\ &= \sum_j Pr(C_j \text{ is satisfied}) \\ &= \sum_j w_j \left(1 - 2^{-l_j}\right) \\ &\geq \frac{1}{2}\sum_j w_j \\ &\geq \frac{1}{2}OPT \end{aligned}$$

An Aside

MAX E3SAT is the version of MAX SAT in which every clause C_j has exactly three variables in it, i.e. $\forall j, l_j = 3$.

A recent theorem due to Hastad says that if there is an approximation algorithm that always returns a solution to the Max E3SAT that is $> \frac{7}{8}OPT$ then $P = NP$.

Note that the simple algorithm on the previous page actually returns an assignment whose expectation is $\geq \frac{7}{8}OPT$ when $\forall j, l_j = 3$. Thus, in some sense, it is a best possible approximation algorithm for MAX E3SAT.