6.842 Randomness and Computation March 3, 2020

Lecture 8

Lecturer: Ronitt Rubinfeld *Scribe: Rogers Epstein*

1 Derandomization via Conditional Expectations

In general, we can think of randomized algorithms $\mathcal{A}$ as a traversal down a binary tree, where the decision to go down the left or right child is determined by our next random bit. We can then think of each leaf as being the end of the algorithm, where an answer is returned. If the algorithm returns a "good" answer with probability p, then a p-fraction of the leaves are "good." For notation, we let m be the number of random bits we use, which is equivalent to the depth of this tree.

This motivates a derandomization approach by instead choosing which child to traverse by choosing one where at least a 1/2-fraction of *its* leaf nodes are "good," assuming that $p \geq \frac{1}{2}$. More formally, let

Definition 1 $p(r_1...r_i) \equiv$ fractions of continuations that end at "good" leaf, using the first $i \leq m$ random bits

In particular,

Definition 2 $p(\Lambda) = Pr[\mathcal{A}$ succeeds on $x] \geq p$, where Λ is the empty list.

Since the next bit is equally likely to be a 0 or 1 after the first i bits have been chosen, we have the following relationship:

$$p(r_1...r_i) = \frac{1}{2}p(r_1...r_i0) + \frac{1}{2}p(r_1...r_i1) \tag{1}$$

Thus, $\exists r_{i+1}$ such that $p(r_1...r_{i+1}) \geq p(r_1...r_i)\ \forall i$. Applying this recursively, we get $\exists r_1...r_m$ such that $p(r_1...r_m) \geq p(r_1...r_{m-1}) \geq ... \geq p(r_1) \geq p(\Lambda) \geq p$. Since $p(r_1...r_m)$ is the probability that when all the bits have been chosen, the algorithm succeeds, it must be either 0 or 1. Since we assume $p \geq \frac{1}{2} > 0$, the above equation implies that $p(r_1...r_m) = 1$!

2 Example: Max Cut

Recall our randomized algorithm for max cut, which expects to return a cut of value $m/2$:

Algorithm 1: MAX CUT

Input : Input graph $G = ([n], E)$
Output: Cut (S, T)

1 Flip $r_1...r_n$
2 $S \leftarrow \{i | r_i = 0\}$
3 $T \leftarrow \{i | r_i = 1\}$
4 Output (S, T)

Now, define the expected cut size conditioned on the first i random bits:

$$e(r_1...r_i) = E_{R_{i+1}...R_n}[|S, T \text{ cut}|;\ r_1...r_i] \tag{2}$$

We've shown before that $e(\Lambda) \geq |E|/2$. We can analyze our approach by defining the following relevant sets at "step" $i + 1$:

$$S_{i+1} \equiv \{\text{nodes } j |\ j \leq i+1, r_j = 0\}$$
$$T_{i+1} \equiv \{\text{nodes } j |\ j \leq i+1, r_j = 1\}$$
$$U_{i+1} \equiv \{\text{nodes } j |\ j \geq i+2\}$$

As we're eventually addings nodes $i+1$ through n to either S or T after this step, we note that

$$e(r_1...r_{i+1}) = |S, T \text{ cut}| + \frac{1}{2}(\#\text{edges touching } U_{i+1}) \tag{3}$$

Our prior observation motivates comparing $e(r_1...r_i0)$ and $e(r_1...r_i1)$. Based on Equation (3), we note that for both these quantities, the size of the cut between S_{i+1} and T_{i+1} remains the same. Also, the edges touching $U_{i+2} = U_{i+1} \backslash \{\text{node } i+1\}$ remains the same, as in either case U_{i+2} and $[n] \backslash U_{i+2}$ are the same. So, we can maximize our eventual cut by putting node $i+1$ on whichever of S_{i+1} and T_{i+1} that maximizes the cut size of S_{i+2} and T_{i+2}. In other words, we can maximize the eventual cut size by putting node $i+1$ on whichever side of the cut that (so far) has fewer edges between it and our node.

Knowing that $e(\Lambda) \geq |E|/2$, we can derandomize this algorithm to guarantee we return a cut of size $\geq |E|/2$, which we know to be a 2-approximation of the optimal cut.

3 Markov Chains

Let Ω be a set of states, and $X_0...X_t \in \Omega$ be a sequence of visited states. A Markov chain is "memoryless", in the sense that the following Markovian property holds:

$$Pr[X_{t+1} = y | X_0 = x_0, X_1 = x_1, ..., X_t = x_t] = Pr[X_{t+1} = y | X_t = x_t] \tag{4}$$

Without loss of generality, we assume transition probabilities are independent of time. In particular, $P(x, y) = Pr[X_{t+1} = y | X_t = x]\ \forall t$, where P is the transition probability matrix for the Markov chain.

One special case of a Markov chain is a random walk on a graph. Here, graphs need not be simple, and often it can be useful to give nodes (multiple) self-loops.

Definition 3 *A* random walk *on a graph $G = (V, E)$ is a sequence of $S_0, S_1, ... \in V$ where S_0 is the start node, and at step i, S_{i+1} is picked uniformly from the out-neighbors of $S_i, N(S_i)$.*

In particular, if $d(i)$ is the out-degree of node i, we have that $P(i, j) = 1/d(i)$ if $(i, j) \in E$ and 0 otherwise. We can then verify that $\forall i$, $\sum_j P(i, j) = 1$. In general, if the rows of P each sum to 1, as is the case for all Markov chains, we call P a *stochastic* matrix. If each column also sums to 1, we call P *doubly stochastic*. Not all Markov chains are doubly stochastic. However, there is a transformation one can make to graphs (as shown in the following lecture) that can make the corresponding matrix doubly stochastic. Additionally, Markov chains for random walks on regular graphs are doubly stochastic.

We further define the notion of being *aperiodic*: $\forall x$, $\gcd\{t : P^t(x, x) > 0\} = 1$. Additionally, a Markov chain is *ergodic* if $\exists t_0$ such that $\forall t \geq t_0$, $P^t(x, y) > 0$. A weaker notion is that of *irreducible*: $\forall x, y\ \exists t \equiv t(x, y)$ such that $P^t(x, y) > 0$. The key difference being that the cutoff of t might depend on the choice of x and y. In fact, ergodic is equivalent to being irreducible and aperiodic.

If our initial distribution is $\Pi^{(0)} = (\Pi_1^{(0)}, ..., \Pi_n^{(0)}$, where $\Pi_i^{(0)} = Pr(\text{start at node } i)$. After t steps, the distribution is given by $\Pi^{(t)} = \Pi^{(0)} P^t$. An important distribution associated to a Markov chain is the stationary distribution:

Definition 4 *We say that Π is a* stationary distribution *if $\forall y$, $\Pi(y) = \sum_x \Pi(x) * P(x, y)$. In other words, $\Pi * P = \Pi$.*

Note that not all Markov chains have stationary distributions (imagine a random walk on a graph which is a single edge, and the initial distribution is concentrated on a single endpoint). Additionally, Markov chains can have multiple stationary distributions (imagine a random walk on a graph with two isolated nodes with self loops).

However, we do have the following important and interesting theorem:

Theorem 5 *Every ergodic Markov chain has a unique stationary distribution.*