6.842 Randomness and Computation February 29, 2012

Lecture 7

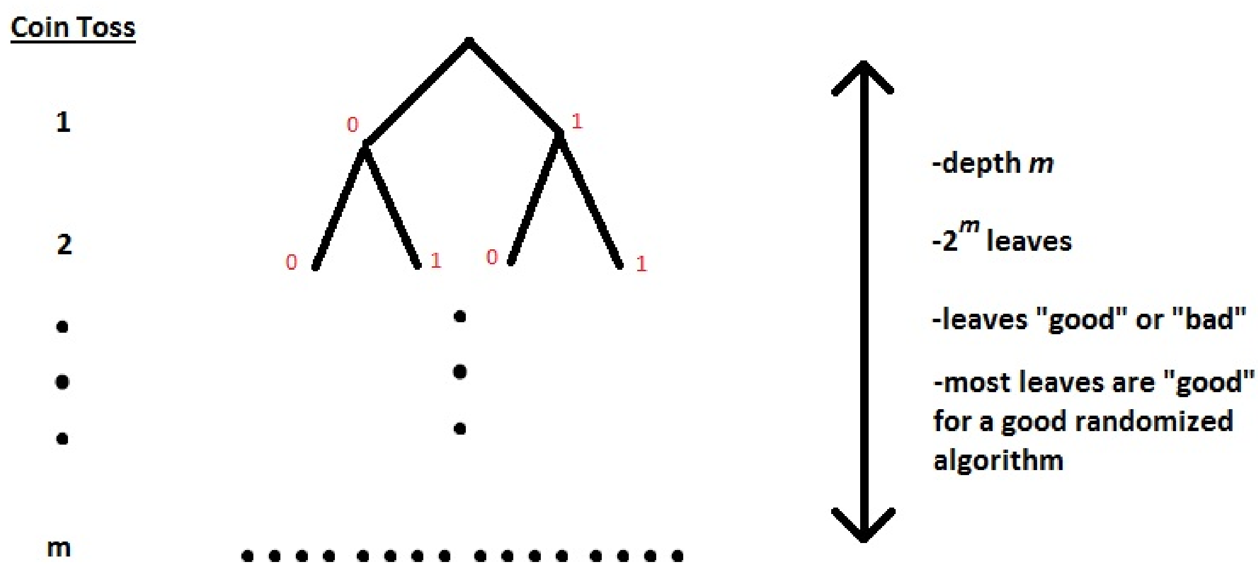
Lecturer: Ronitt Rubinfeld *Scribe: John Ruszczynski*

1 Overview

- Derandomization via the Method of Conditional Expectations
  - Application to Max-Cut
- Markov Chains
  - Definitions
  - Random walk on graphs
  - Cover time

2 Method of Conditional Expectations

Essentially, this is a "guided search for a good random string." At each node, we pick the next node such that the probability of reaching a "good" final leaf does not decrease.



"Good" leaves correspond to sequences of randomized bits that result in the randomized algorithm being correct (accepting if the input $x \in L$, rejecting otherwise). For $L \in BPP$, this means that the majority of leaves are "good," since the algorithm is correct with probability at least $\frac{2}{3}$.

2.1 The Procedure

Fix randomized algorithm A, input x. Let $m \equiv \#$ of random bits used by A on x.

For $1 \le i \le m$:
  $r_1, \ldots, r_i \in \{0,1\}$ (random bits already chosen)
  Let $p(r_1, \ldots, r_i) \equiv$ fraction of continuations leading to a good leaf $= \frac{1}{2}p(r_1, \ldots, r_i, 0) + \frac{1}{2}p(r_1, \ldots, r_i, 1)$.
  [Note: $p(\Lambda) \ge \frac{2}{3}$ for a "good" randomized algorithm for a language in BPP.]
  By averaging, $\exists r_{i+1}$ s.t. $p(r_1, \ldots, r_i, r_{i+1}) \ge p(r_1, \ldots, r_i)$.
  $(\star)$ Pick r_{i+1} that maximizes $p(r_1, \ldots, r_{i+1})$.

This procedure gives: $p(r_1, \ldots, r_m) \ge \cdots \ge p(r_1, r_2) \ge p(r_1) \ge p(\Lambda) = \frac{2}{3}$. Since $p(r_1, \ldots, r_m) \in \{0,1\}$ and $p(r_1, \ldots, r_m) \ge \frac{2}{3} \Rightarrow p(r_1, \ldots, r_m) = 1$, so the procedure finds a good leaf.

Question: how do we implement $(\star)$? Note that it is often sufficient for the p's for each continuation to be approximated. Also we have not yet made use of $p(\Lambda) \ge \frac{2}{3}$ beyond the weaker condition $p(\Lambda) > 0$. However having $p(\Lambda)$ be greater than a constant makes implementing $(\star)$ easier.

2.2 Application to Max-Cut

Recall the randomized algorithm:
  Pick $r_1, \ldots, r_n \in \{0,1\}$
  Put node i in S if $r_i = 0$; put it in T if $r_i = 1$.
  Output S, T.

Derandomization:
  Let $e(r_1, \ldots, r_i) = E_{R_{i+1}, \ldots, R_n}[\ |\text{cut}(S,T)|\ |$ choices of $r_1, \ldots, r_i$ made so far].
  $e(\Lambda) = \frac{m}{2}$ ($m = \#$ edges in the graph).

  To calculate $e(r_1, \ldots, r_{i+1})$, let:
    $S_{i+1} = \{j | j \le i+1, r_j = 0\}$
    $T_{i+1} = \{j | j \le i+1, r_j = 1\}$
    $U_{i+1} = \{j | j \ge i+2\}$ ("undecided").

$$\text{So } e(r_1, \ldots, r_{i+1}) = \overbrace{(\#\text{ edges between } S_{i+1} \text{ and } T_{i+1})}^{\text{Term I}} + \overbrace{\frac{1}{2}(\#\text{ edges touching } U_{i+1})}^{\text{Term II}}.$$

We do not actually need to calculate $e(r_1, \ldots, r_{i+1})$ for the procedure; we need only determine whether $e(r_1, \ldots, r_i, 0) \overset{?}{\lesseqgtr} e(r_1, \ldots, r_i, 1)$. Note that Term II is the same for both choices of r_{i+1} and that Term I differs only in edges adjacent to the $(i+1)^{\text{st}}$ node. Thus to maximize $e(r_1, \ldots, r_i)$, maximize Term I according to max{# edges between node $i+1$ and S_{i+1}, # edges between node $i+1$ and T_{i+1}}.

This corresponds to the greedy algorithm for Max-Cut:
  $S, T \leftarrow \emptyset$
  For $i = 0, \ldots, n-1$:
    If # edges between node $i+1$ and $T >$ # edges between node $i+1$ and S
      Set $S \leftarrow S \cup \{i+1\}$
    Else set $T \leftarrow T \cup \{i+1\}$

3 Markov Chains

Definition: Given Ω = ground set. (For this class, Ω will always be finite. Can think of it as the nodes in a graph or states of a finite system.) A **Markov chain** is a finite sequence of random variables, $X_0, \ldots, X_t \in \Omega$ such that:

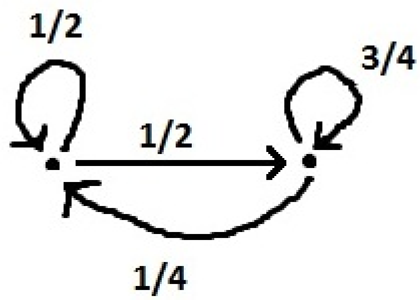
$$\forall t, \forall x_0, \ldots, x_t, y \in \Omega, \mathbb{P}[X_{t+1} = y | X_0 = x_0, \ldots, X_t = x_t] = \mathbb{P}[X_{t+1} = y | X_t = x_t] \quad (1)$$

Note that (1) is known as the **Markovian Property** and is also referred to as "memorylessness." It says that the next state in a Markov chain only depends on the current state.

For this course, we will assume that transition probabilities are independent of time, so $\forall t, \forall x, y \in \Omega$, $\mathbb{P}[X_{t+1} = y | X_t = x] = \mathbb{P}[X_1 = y | X_0 = x]$.

3.1 Representations

- Completed directed graph (with self loops)



  Note that the outgoing edges sum to 1, but they are otherwise arbitrary nonnegative numbers.

- Transition Matrix

  $P(x, y) = \mathbb{P}[X_{t+1} = y | X_t = x]$.

  The probability distribution of states at a given time t is represented by a vector π_t. π_0 is the *start state* and it can be a single state (e.g. $\pi_0 = (1, 0)$) or a distribution ($\pi_0 = (1/2, 1/2)$).

  The transition matrix corresponding to the above graph is $P = \begin{pmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{pmatrix}$

  Given an initial distribution π_0, the distributions at time t is given by:

  $\pi_1 = \pi_0 P$
  $\pi_2 = \pi_1 P = (\pi_0 P)P = \pi_0 P^2$
  $\vdots$
  $\pi_t = \pi_0 P^t$.

  Example: In the graph above, start in state 1. Then at $t = 1$, the distribution is $\pi_1 = \pi_0 P = (1, 0) \begin{pmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{pmatrix} = (1/2, 1/2)$.

Definition: A **stationary distribution** π satisfies $\pi = \pi P$.
Question: When does π exist? When is it unique?

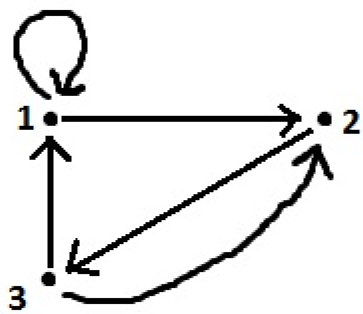
Definition: An **ergodic Markov chain** is one where $\exists t_0$ such that $\forall t > t_0, \forall x, y \in \Omega, P^t(x,y) > 0$.

Theorem: For Ω finite, a Markov chain is ergodic *if and only if*

1. it is irreducible ($\forall x, y\ \exists t_{x,y}$ such that $P^{t_{x,y}}(x,y) > 0$).
2. it is aperiodic ($\forall x \in \Omega, \gcd\{t | P^t(x,x) > 0\} = 1$).

3.2 Random Walks on Graphs

- At each step, go to a uniformly chosen neighbor.
- Can have multi-edges and self-loops.



For the above graph, the transition matrix is $P = \begin{pmatrix} 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \\ 1/2 & 1/2 & 0 \end{pmatrix}$.

In general the transition matrix is given by $P_{ij} = \frac{\#\ (i,j)\ \in\ E}{\text{out degree of } i}$.

Note that this implies that $\sum_j P_{ij} = 1\ \forall i$. Such a matrix is called a stochastic matrix.

Definition: A **stochastic matrix** is one where all rows sum to 1.

Definition: A **doubly stochastic matrix** is one where all rows and all columns sum to 1.

Examples of doubly stochastic matrices are the transition matrices for undirected graphs and graphs with the same in degree and out degree at every node ($d_{in} = d_{out} = d\ \forall$ nodes).

Fact: for such matrices, the stationary distribution is given by $\pi = (\frac{\deg(x_1)}{2^m}, \frac{\deg(x_2)}{2^m}, \ldots)$. Thus for a d-regular graph, the stationary distribution is uniform.