22C:199 Applications of Chernoff bounds and an Introduction to Random Walks

Scribe: Ganesh Venkataraman

15th October 2003

In the previous lecture, we had proved the following lemma:

Lemma 1 *Suppose H is a directed graph with no parallel edges and the edges have a min-degree x and max-degree y, where $x \geq 1000$ and $y \leq 4x$, then the vertices of H can be colored red and blue such that for every $v \in V(H)$, the number of red out-neighbors of v is in $[\delta^+(v)/2 - (\delta^+(v))^{2/3}, \delta^+(v)/2 + (\delta^+(v))^{2/3}]$ and the number of red in-neighbors of v is in $[\delta^-(v)/2 - (\delta^-(v))^{2/3}, \delta^-(v)/2 + (\delta^-(v))^{2/3}]$. Similar result is true for the blue in-neighbors and out-neighbors as well.*

We now need to prove the following:

Lemma 2 *Let G be a directed graph with no parallel edges and min-vertex degree $\geq k \geq 1$ and max-degree $\leq 2k$. Then the vertices of G can be colored with $\frac{k}{2^{16}}$ colors, each used so that for each color, the induced subgraph has vertex degree in $[a, 4a]$, $a \leq 1$.*

Proof: The basic idea is to repeatedly use lemma 2 to get the coloring of $\frac{k}{2^{16}}$ colors. We apply lemma 2 r times where $r = \lfloor log_2 k \rfloor - c$. So we get a total of $2^r = 2^{\lfloor log_2 k \rfloor - c} \geq \frac{k}{2^{16}}$ colors. Assume $c = 15$ and $k \geq 2^{16}$. This implies $r \geq 1$. Now, we first check if lemma 2 is valid for the very first time. In the first step, min-degree $= k \geq 2^{16} > 1000$ and max-degree $\leq 2k < 2k$ and hence the lemma holds the first time. Now, let $f(x) = \frac{1}{2}x - x^{2/3}$ and $g(x) = \frac{1}{2}x + x^{2/3}$. Let $z \geq k$ represent the min-degree. Let $x_0 = z$ and $x_{i+1} = f(x_i)$ for $i = 1, 2, \ldots$. Then using some calculations we can show that:

$$x_j \geq \frac{2}{3}(2^{-j}z) \forall j = 1, 2, \ldots r \tag{1}$$

Let z' represent the max-degree. Let $y_{i+1} = g(y_i)$ for $i = 1, 2, \ldots$. We can show that

$$y_j \leq \frac{4}{3}(2^{-j}z') \forall j = 1, 2, \ldots r \tag{2}$$

Hence, from the above two bounds, we conclude that:

$$y_j \leq \frac{4}{3}(2^{-j} * 2k) \leq 4 * \frac{2}{3}(2^{-j}z) \tag{3}$$

$$\Rightarrow y_j \leq 4x_j \tag{4}$$

Hence it is easy to see that for the choice of $c = 15$, the min-degree and max-degree requirements are satisfied at every level and hence the lemma holds. □

Random Walks

Let $G = (V, E)$ be an undirected graph. Let $v_0 \in V$ be chosen arbitrarily as a source of our walk. Random walk is a sequence of vertices $v_0, v_1, \ldots$ where $v_i, i \geq 1$ is chosen from the neighbors of v_{i-1}, uniformly at random independent of all previous choices. (This is a specific case of a random walk). Now we could ask the following questions above the random walk:

1. What is the expected time for the walk to visit all the vertices in G?
2. Given a particular vertex (say u), what is the expected time to reach u the first time?

Markov's chains are used to answer the above questions related to random walks.

Example of Random Walk

Let $G = K_n$. Let $v_0 \in V(G)$ be a source and let $v \neq v_0$ be an arbitrary vertex in G. What is the expected time by which a simple random walk first visits v. This problem can be solved without using Markov's chain and the number of steps is given by the summation $\frac{1}{n-1} + \frac{2(n-2)}{(n-1)^2} + \frac{3(n-2)(n-3)}{(n-1)^3} + \ldots$.

Solving 2-SAT using Random Walk

To provide some more motivation for, we now present an algorithm for solving 2-SAT using random walks. Recall that k-SAT is NP-hard for $k \geq 3$, but polynomial time solvable for $k = 2$.

Randomized algorithm for 2-SAT

1. Start with an arbitrary truth assignment
2. If there is an unsatisfactory clause, pick one unsatisfactory clause arbitrarily
3. Pick one of the two literals in this clause uniformly at random and complement it's value
4. Go back to step (2)

Let us assume that the given instance of 2-SAT has a satisfying truth assignment, A. Let T be the current truth assignment. Define correctness(T) as the number of variables that have the same value in A and T. Now it $0 \leq correctness(T) \leq n$ where n is the number of variables. At each step correctness(T) increases by 1 with probability 1/2. Later, we show that the number of steps needed for correctness(T) to reach it's final values is $O(n^2)$. So we wait for $2cn^2$ steps and then decide whether the given instance is satisfiable or not. Note that this algorithm has a one sided error, since it can never produce declare an instance as satisfiable if it does not have a satisfiable assignment. It could declare a satisfiable assignment as unsatisfiable with finite probability. However, this probability (as we will see later) is quite low.