CMPUT 675: Randomized Algorithms **Fall 2005**

Lecture 15: Oct 27

Lecturer: Mohammad R. Salavatipour *Scribe: Zac Friggstad*

15.1 Example 3: Sampling Colorings

Given a graph with n vertices where each vertex is of degree $\leq \Delta$, we want to sample uniformly at random from a k-coloring where $k \geq 4\Delta + 1$. Note that it is hard to approximate the *chromatic number* $\chi(G)$ (minimum number of colors required to color a graph G), within a factor of $O(n^{1-\epsilon})$ for any $\epsilon > 0$, but a coloring with $\Delta + 1$ colors is easy to construct (use greedy algorithm).

greedy_color$(G(V, E))$
  for each $v \in V$ **do**
    pick one of the $\Delta + 1$ colors c such that no neighbor of v is already colored c
    assign color c to v

For each vertex v there is always at least one color c to assign to v in this algorithm since there are at most Δ neighbors. Therefore we can easily construct a coloring with $\Delta + 1$ colors so we can also construct a coloring with $4\Delta + 1$ colors.

Consider the following process. Pick a vertex $v \in V$ and color $c \in \{1 \ldots k\}$ uniformly at random and change the color of v to c if this results in a valid coloring, otherwise do nothing. The claim is that this is a rapidly mixing Markov chain where the states are all possible $k = 4\Delta + 1$ colorings. Since there are k colors and n vertices then there can be at most k^n ways to assign colors to vertices so this Markov chain is finite. It is ergodic because if a vertex v and color c are chosen such that v is already colored c, then no change occurs.

Finally, this Markov chain is irriducible. Say we are trying to get from one coloring C_1 to another C_2. Consider a vertex v that is colored differently among the two colorings. If the color of v can be changed to its color in C_2 immediately, then there is no problem. However, if a conflict occurs then it must be that an adjacent vertex w has the color c of v in the coloring C_1. Vertex w cannot be colored the same in C_1 and C_2 since the vertex v is colored c in C_2. We can pick a color $c' \neq c$ for w such that this color is not a color of any adjacent vertices to w in C_1, which is possible since $k \geq 4\Delta + 1$. Change the color of w to c'. Repeat this for all neighbors of v that have color c. After this, vertex v can then be colored c with no conflictions resulting in one more vertex of having the same color in the current coloring and the destination coloring C_2. Repeat until all vertices are colored the same.

To see that this is rapidly mixing, consider the following simple coupling $Z = (X, Y)$. Choose the same vertex v and color c in both uniformly at random. Let d_t be the number of vertices with different colors in X and Y. By the nature of the Markov chain, $|d_t - d_{t+1}| \leq 1$. Consider the following probabilities.

- Since there are $\frac{d_t}{n}$ different ways to choose one of the vertices with different colors and at most 2Δ different colors among the neighbors of a vertex in both chains X and Y, then

$$\Pr(d_{t+1} = d_t - 1 \,|\, d_t > 0) \geq \frac{d_t}{n} \cdot \frac{k - 2\Delta}{k}.$$

- If a vertex v is colored the same in both chains X and Y before a step but is colored differently after the step, then v must have a neighbor w that does not have the same color in both chains since some neighbor interfered with an attempted color change in one chain but not in the other. Every vertex colored differently in the two chains can affect at most Δ neighbors in this way, thus there are at most $\frac{d_t \Delta}{n}$ ways that this can happen with a $\frac{1}{k}$ chance of chosing this interfering color, so

$$\Pr(d_{t+1} = d_t + 1 \,|d_t > 0) \leq \frac{d_t \Delta}{n} \cdot \frac{2}{k}$$

where the 2 accounts for the symmetry of this case.

So

$$\begin{aligned}
\mathrm{E}[d_{t+1}\,|d_t] &= d_t + \Pr(d_{t+1} = d_t + 1) - \Pr(d_{t+1} = d_t - 1) \\
&\leq d_t + \frac{2 d_t \Delta}{nk} - \frac{d_t}{n} \cdot \frac{k - 2\Delta}{k} \\
&\leq d_t \left(1 - \frac{k - 4\Delta}{nk}\right)
\end{aligned}$$

and, by using the conditional expectation and induction as in example 2 from the previous lecture,

$$\begin{aligned}
\mathrm{E}[d_t] &\leq d_0 \left(1 - \frac{k - 4\Delta}{nk}\right)^t \\
&\leq n \left(1 - \frac{k - 4\Delta}{nk}\right)^t \\
&\leq n e^{-t(k-4\Delta)/kn}
\end{aligned}$$

If

$$t = \frac{nk}{k - 4\Delta} \ln\left(\frac{n}{\varepsilon}\right),$$

then $d_t \leq \varepsilon$, which results in

$$\tau(\varepsilon) = \frac{nk}{k - 4\Delta} \ln\left(\frac{n}{\varepsilon}\right)$$

which is polynomial in n and in $\ln\left(\frac{1}{\varepsilon}\right)$ and exhibits that this Markov chain is rapidly mixing. As a note, the current best result for sampling from random k colorings is for $k \geq \frac{11}{6}\Delta$.

15.2 Shönig's $3SAT$ algorithm

Given a Boolean expression ϕ in 3CNF with m clauses and n variables, determining if ϕ is satisfiable is an **NP**-hard problem. A simple exhaustive search algorithm takes $O((n+m)2^n)$ time where the 2^n truth assignments are generated in $O(n)$ time and the truth of all m clauses must be verified for each of these truth assignments. Instead, consider the following random process. Start with an arbitrary truth assignment and at each step of the random process, choose an unsatisfied clause and a flip the truth value of one of the variables chosen uniformly at random from the clause.

In the analysis, let A^* be a specific satisfying truth assignment of ϕ. Since the truth value of at least one variable in an unsatisfied clause differs from A^* (otherwise the clause is satisfied), then the probability of getting one step closer to A^* is at least $1/3$ while the probability of moving away from A^* is at most $2/3$. Being pessimistic, we will say that the probability of moving closer is exactly $1/3$ whereas the probability of moving away is exactly 2/3. An upper bound on the expected number of steps in this Markov chain is also an upper bound on the original Markov chain.

Imagining that k variables differ from A^*, then the probability of going straight to the assignment A^* is at least $(1/3)^k$. If k is about $n/2$ then with probability at least $(1/3)^{n/2}$ we get to n in $n/2$ steps. Repeating this process $2 \cdot 3^{n/2}$ times, the probability of getting to A^* is at least 1/2. This algorithm has about $O(3^{n/2})$ or $O((1.78)^n)$ iterations which is significantly better than the naive search. We improve upon this algorithm using two ideas:

- Instead of repeating k steps, we repeat $3k$ steps and show that the probability of getting to A^* is at least $(\frac{1}{2})^k$.
- We consider all possible values of k (not just $n/2$) for the initial assignment.

Consider the modified version of this algorithm.

```
random_3-SAT(φ)
    pick a truth assignment A to φ uniformly at random
    repeat at most t times
        if φ is satisfied with A, then return A
        else pick an unsatisfied clause c_i uniformly at random and flip the truth
             value in A of one of the variables chosen uniformly at random from c_i.
    return φ is unsatisfiable
```

Say in one of these random walks where k values are incorrect, t steps were taken. If i steps were to the left and $k+i$ steps were to the right, then $t = k + 2i$ steps were taken in total. This happens with probability

$$\binom{k+2i}{i}\left(\frac{2}{3}\right)^i\left(\frac{1}{3}\right)^{k+i}.$$

If we consider this expression at $i = k$, then the value is

$$\binom{3k}{k}\left(\frac{1}{3}\right)^{2k}\left(\frac{2}{3}\right)^k$$

which provides a lower bound for the probability that the algorithm reaches A^*. Thus,

$$\begin{aligned}
\binom{3k}{k}\left(\frac{1}{3}\right)^{2k}\left(\frac{2}{3}\right)^k &= \frac{(3k)!}{k!(2k)!}\left(\frac{1}{3}\right)^{2k}\left(\frac{2}{3}\right)^k \\
&\geq \frac{\sqrt{2\pi(3k)}}{4\sqrt{2\pi k}\sqrt{2\pi(2k)}}\left(\frac{3k}{e}\right)^{3k}\left(\frac{e}{2k}\right)^{2k}\left(\frac{e}{k}\right)^k\left(\frac{1}{3}\right)^{2k}\left(\frac{2}{3}\right)^k \\
&= \frac{\sqrt{3}}{8\sqrt{\pi k}}\left(\frac{27}{4}\right)^k\left(\frac{2}{27}\right)^k \\
&= \frac{\sqrt{3}}{8\sqrt{\pi k}}\left(\frac{1}{2}\right)^k
\end{aligned}$$

Lemma 15.1 *For $t = 3n$, the probability that after t steps, we find a satisfying truth assignment is at least*

$$\frac{c}{\sqrt{k}}\left(\frac{3}{4}\right)^n$$

where $c = \sqrt{3}/8\sqrt{\pi}$.

Proof:

$$\begin{aligned}
\Pr(\phi\ satisfied) &\geq \sum_{k=0}^{n} \binom{n}{k} \frac{1}{2^n} \frac{c}{\sqrt{k}} \frac{1}{2^k} \\
&\geq \frac{c}{\sqrt{n}2^n} \sum_{k=0}^{n} \binom{n}{k} \frac{1}{2^k} \\
&= \frac{c}{\sqrt{n}2^n} \left(\frac{3}{2}\right)^n \\
&= \frac{c}{\sqrt{n}} \left(\frac{3}{4}\right)^n
\end{aligned}$$

where the summing of the series is justified by

$$\sum_{k=0}^{n} \binom{n}{k} \left(\frac{1}{2}\right)^k (1)^{n-k} = \left(\frac{1}{2}+1\right)^n$$

by the binomial theorem. ■

If this algorithm is repeated $(4/3)^n 2\sqrt{n}/c$ times, then the probability of success is at least $1/2$. The total number of iterations is $O(\sqrt{n}\ (4/3)^n)$ or about $(\sqrt{n}\ (1.33)^n)$.