CSE 200 Computability and Complexity Monday, May 20, 2013

Lecture 15: Randomized Computation (cont.)

Instructor: Professor Shachar Lovett *Scribe: Dongcai Shen*

1 Random Walk Algorithms for k-SAT

1.1 A random walk algorithm for 2-SAT

2-SAT. $\phi(\mathbf{x}) = (x_1 \vee \neg x_2) \wedge (x_3 \vee x_1) \wedge \cdots$. ϕ is satisfiable if there exists an assignment a s.t. $\phi(a) = 1$.

Algorithm 1 Random Walk for 2-SAT [2]

1: Choose $r \in \{0,1\}^n$ randomly.
2: If $\phi(r) = 1$, done.
3: otherwise, exists clause (e.g.) $C = x_i \vee x_j$ for which $r_i = r_j = 0$.
4: choose either i, j randomly, flip r_i or r_j.
5: **Go to** 2

Theorem 1 (Papadimitriou [2]) *If ϕ is satisfiable, then whp after $O(n^2)$ steps r reaches a satisfiable assignment.* □

Proof Fix a satisfying assignment a. Define the distance of r to a, denoted $\mathsf{dist}(r, a)$, as the number of coordinates where $r_i \neq a_i$ (this is called *Hamming distance*). Let $r = r^1, r^2, r^3, \cdots$ be the assignments generated by the algorithm, and let $d_i = \mathsf{dist}(r^i, a)$. Note that as we only change one bit of the assignment, $d_{i+1} = d_i + \Delta_i$ where $\Delta_i \in \{-1, 1\}$. We claim that $\Pr[\Delta_i = -1] \geq 1/2$. Once this is established, we have a random walk of $d_1, d_2, \ldots$ on the range $\{0, 1, 2, \ldots, n\}$, which decreases in each step with probability at least $1/2$. It can be shown that such a random walk will reach 0 after $O(n^2)$ steps whp.

To finish, we need to prove that $\Pr[b_i = -1] \geq 1/2$. To see that, assume the value of r^i on the current two bits is (α_1, α_2) and the value of a is (β_1, β_2), where $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \{0, 1\}$ and $(\alpha_1, \alpha_2) \neq (\beta_1, \beta_2)$ since r^i does not satisfy ϕ. Then, it can be verified that

- If $\alpha_1 = \beta_1, \alpha_2 \neq \beta_2$ then $\Pr[b_i = -1] = 1/2$.
- If $\alpha_1 \neq \beta_1, \alpha_2 = \beta_2$ then $\Pr[b_i = -1] = 1/2$.
- If $\alpha_1 \neq \beta_1, \alpha_2 \neq \beta_2$ then $\Pr[b_i = -1] = 1$.

■

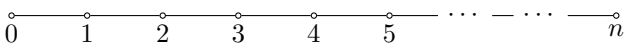

Figure 1: Random walk

This algorithm runs in $O(n^2)$ expectation time.

1.2 A random walk algorithm for 3-SAT

What about 3-SAT? Same algorithm. $r \in \{0,1\}^n$ randomly. If $\phi(r) \neq 1$, find an arbitrary clause C and flip one of the variables of r in C. $\mathbf{a} = (0,0,1) \to \mathbf{r} = (0,1,1)$.

$$\mathsf{dist} \text{ time } t+1 = \mathsf{dist} \text{ time } t + \begin{cases} -1 & \text{w. prob. } \frac{1}{3} \\ +1 & \text{w. prob. } \frac{2}{3} \end{cases} \tag{1}$$

With some careful analysis, it resolves to $(\frac{4}{3})^n$. k-SAT's such algorithm runs in time $2^{n(1-O(\frac{1}{k}))}$. This is the best known algorithm for k-SATup to the constants in the $O(\cdot)$.

2 List Coloring

Definition 2 (A simple path in graphs)

- Input: An undirected graph $G = (V, E)$ and a number k.
- Question: Does G have a simple path of length k. A *simple path* is a sequence of vertices $v_1, v_2, \cdots, v_k \in V$ s.t. $(v_i, v_{i+1}) \in E \ \forall i \in [k-1]$ and $v_i \neq v_j \ \forall i \neq j$. □

Algorithm 2 A trivial algorithm

1: Try all combinations.

Algorithm 2's running time is $O(n^k)$. The following randomized algorithm (Algorithm 2) runs in time $n^{O(1)} \cdot 2^k$.

Let $\chi : V \to [k]$ be a random coloring. Let $p \stackrel{\text{def}}{=} v_1 v_2 \cdots v_k$ be a simple path in G. We say χ is *good* if $\chi(v_1), \chi(v_2), \cdots, \chi(v_k)$ are all different (e.g., take all k possible colors).

Claim 3 $\mathbf{Pr}[\chi \text{ is good}] \approx \mathrm{e}^{-k}$. □

Claim 4 *If χ is good, we can find a simple path in time $n^{O(1)} \cdot 2^k$.* □

Proof of 3:

$$\mathbf{Pr}[v_1, \cdots, v_k \text{ get all } k \text{ different colors}] = \frac{k!}{k^k} \approx \frac{(\frac{k}{\mathrm{e}})^k}{k^k} = \left(\frac{1}{\mathrm{e}}\right)^k$$

where Stirling approximation [3] was applied. ■

Proof of 4: For $A \subset [k]$, define

$$S_A \stackrel{\text{def}}{=} \{v \in V : \text{there is a simple path of length } |A|, \text{ ends in } v \text{ and takes colors in } A\}.$$

We will compute S_A using dynamic programming, and then the result can be derived from $S_{[k]}$.

Say we want to compute S_A with $|A| = a$. We know already $S_{A'}$ for all sets A' of size $|A'| < a$. Let $A \stackrel{\text{def}}{=} \{c_1, \cdots, c_a\}$. A vertex $v \in S_A$ of color $c \in A$ iff v is a neighbor of a vertex $u \in S_{A \setminus c}$. Hence we can compute

$$S_A = \bigcup_{c \in A} \left\{ v : \chi(v) = c,\ v \text{ neighbor of } u \in S_{A \setminus \{c\}} \right\}.$$

Computing S_A from $\{S_{A \setminus \{a\}}\}$ takes time $n^{O(1)}$. So, to compute $S_{[k]}$ takes $2^k \cdot n^{O(1)}$ time. ■

3 Relations Between Randomized Complexity Classes and Other Complexity Classes

Theorem 5 (Adelman [1]) $\mathsf{BPP} \subseteq \mathsf{P/poly}$. □

Proof Suppose a language $L \in \mathsf{BPP}$. There is a poly-time machine $M(x, r)$ s.t.

- $x \in L \Rightarrow \mathbf{Pr}_r[M(x) = 1] \geq 2/3$.
- $x \notin L \Rightarrow \mathbf{Pr}_r[M(x) = 1] \leq 2/3$. □

By amplification (majority of $n^{O(1)}$ independent runs), we get a new TM M' such that

$$\mathop{\mathbf{Pr}}_{r}[M'(x, r) = L(x)] > 1 - 2^{-2n}.$$

So there exists a fixing of the randomness r^* such that

$$\mathop{\mathbf{Pr}}_{x \in \{0,1\}^n}[M'(x, r^*) = L(x)] \geq 1 - 2^{-2n} \Rightarrow M'(x, r^*) = L(x) \quad \forall x \in \{0,1\}^n$$

Let $C(x) \stackrel{\text{def}}{=} M'(x, r^*)$ where r^* is hard-wired to C. ■

Question 6 $\mathsf{BPP} \subseteq \mathsf{NP}$*?* □

Theorem 7 (Sipser-Gács [4]) $\mathsf{BPP} \subseteq \Sigma_2 \cap \Pi_2$. □

Proof If suffices to prove that $\mathsf{BPP} \subseteq \Sigma_2$. Apply a similar argument at the beginning of the proof of Theorem 5, Assume there exists a TM M s.t. $\mathbf{Pr}_r[M(x, r) = L(x)] > 1 - 2^{-2n}$.

Let $S_x \stackrel{\text{def}}{=} \{r \in \{0,1\}^m : M'(x, r) = 1\}$ where $m = n^{O(1)}$. Then

- $x \in L \Rightarrow |S_x| > (1 - 2^{-2n}) \cdot 2^m$.
- $x \notin L \Rightarrow |S_x| < 2^{-2n} \cdot 2^m$. □

Claim 8 *Set* $k = \frac{m}{n}$.

(1) If $x \in L$*, there exists* $y_1 \cdots y_k \in \{0,1\}^m$ s.t. $\cup_{i=1}^k (S_x + y_i) = \{0,1\}^m$.

(2) If $x \notin L$*, for every* $y_1 \cdots y_k \in \{0,1\}^m$, $\cup_{i=1}^k (S_x + y_i) \neq \{0,1\}^m$. □

□

Proof of Claim 8:

(2) $x \notin L$, $|S_x| < 2^{m-2n}$. $|\cup_{i=1}^k (S_x + y_i)| \leq \sum_{i=1}^k |S_x + y_i| \leq k \cdot |S_x| \leq k \cdot 2^{m-2n} < 2^m$.

(1) $x \in L$, $|S_x| > 2^m(1 - 2^{2n})$. Choose $y_1, \cdots, y_k \in \{0,1\}^m$ randomly. Fix $z \in \{0,1\}^m$.

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{y_1, \cdots, y_k}\left[z \notin \cup_{i=1}^k (S_x + y_i)\right] \\
=&\mathbf{Pr}\left[z \notin y_1 + S_x, z \notin y_2 + S_x, \cdots, z \notin y_k + S_x\right] \\
=&\mathop{\mathbf{Pr}}_{y_1, \cdots, y_k}[y_1, \cdots, y_k \notin S_x + z] \\
\leq&\left(\frac{2^m - |S_x + z|}{2^m}\right)^k \leq 2^{-2nk} = 2^{-2m}.
\end{aligned}$$

Hence,

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{y_1, \cdots, y_k}\left[\exists z \in \{0,1\}^m, z \notin \cup_{i-1}^k (S_x + y_i)\right] \\
\leq&\sum_{z \in \{0,1\}^m} \mathop{\mathbf{Pr}}_{y_1, \cdots, y_k}[z \notin \cup (S_x + y_i)] \\
\leq&2^m \cdot 2^{-2m} = 2^{-m}.
\end{aligned}$$

■ Now,

$$x \in L \Leftrightarrow \exists y_1, \cdots, y_k \in \{0,1\}^m, \forall z \in \{0,1\}^m, \bigvee_{i=1}^{k} \underbrace{M(x, y_i + z) = 1}_{y_i + z \in S_x}$$

Therefore, $L \in \Sigma_2$.

■

4 Probabilistic Constructions

Probabilistic constructions are very useful to show the existence of various combinatorial structures. We will illustrate this with codes. An (n, k, d) binary code is a subset $C \subseteq \{0,1\}^n$ with $|C| = 2^k$ where for any $x \neq y \in C$, $\mathsf{dist}(x, y) \geq d$.

A code is considered *good* if $k = \alpha n$, $d = \beta n$, for some constants $\alpha, \beta > 0$. We will prove good codes exist by a simple probabilistic argument.

Theorem 9 *For any $\alpha, \beta > 0$, small enough, there exists (n, $k \stackrel{\text{def}}{=} \alpha n$, $d \stackrel{\text{def}}{=} \beta n$) codes for all large enough n.*

Proof Let $C \subset \{0,1\}^n$ of size $|C| = 2^k$ be chosen uniformly. Then

$$\begin{aligned}
&\mathbf{Pr}\left[\exists x, y \in C, \mathrm{d}(x, y) \leq d\right] \\
\leq &\sum_{\substack{x,y \in \{0,1\}^n \\ \mathsf{dist}(x,y) \leq d}} \mathbf{Pr}\left[x, y \in C\right] \\
\leq &2^n \sum_{i=0}^{d} \binom{n}{i} \left(\frac{2^k}{2^n}\right)^2 && k = \alpha n, d = \beta n, H(\alpha) = \alpha \log \frac{1}{\alpha} (1-\alpha) \log \frac{1}{1-\beta} \\
= &2^n \cdot 2^{(H(\alpha)+o(1))\cdot n} \cdot 2^{2\beta - 2n} \\
= &2^{(H(\alpha)+2\beta+o(1)-1)n}
\end{aligned}$$

When α, β are small enough, $H(\alpha) + 2\beta < 1$, almost all (n, k, d) codes are good (for large enough n). ■

References

[1] Leonard M. Adleman. Two theorems on random polynomial time. In *FOCS*, pages 75–83, 1978.

[2] Christos H. Papadimitriou. On selecting a satisfying truth assignment (extended abstract). In *FOCS*, pages 163–169, 1991.

[3] Herbert Robbins. A remark on stirling's formula. *The American Mathematical Monthly*, 62(1):pp. 26–29, 1955.

[4] Michael Sipser. A complexity theoretic approach to randomness. In *STOC*, pages 330–335, 1983.