6.841: Advanced Complexity Theory Fall 2012

Lecture 11 — October 16, 2012

Prof. Dana Moshkovitz *Scribe: Hyuk Jun Kweon*

1 Overview

In the previous lecture, there was a question about whether or not true randomness exists in the universe. This question is still seriously debated by many physicists and philosophers. However, even if true randomness does not exist, it is possible to use pseudorandomness for practical purposes. In any case, randomization is a powerful tool in algorithms design, sometimes leading to more elegant or faster ways of solving problems than known deterministic methods.

This naturally leads one to wonder whether randomization can solve problems outside of **P**, deterministic polynomial time. One way of viewing randomized algorithms is that for each possible setting of the random bits used, a different strategy is pursued by the algorithm. A property of randomized algorithms that succeed with bounded error is that the vast majority of strategies succeed. With error reduction, all but exponentially many strategies will succeed. A major theme of theoretical computer science over the last 20 years has been to turn algorithms with this property into algorithms that deterministically find a winning strategy. From the outset, this seems like a daunting task – without randomness, how can one efficiently search for such strategies? As we'll see in later lectures, this is indeed possible, modulo some hardness assumptions.

There are two main objectives in this lecture. The first objective is to introduce some probabilistic complexity classes, by bounding time or space. These classes are probabilistic analogues of deterministic complexity classes **P** and **L**. The second objective is to prove that the undirected connectivity problem is in the certain probabilistic complexity class **RL**, by intorducing a space-efficient algorithm for solving it.

2 Probabilistic Complexity Classes

The aim of this section is to introduce several probabilistic complexity classes. These classes are obtained by bounding time or space of a probabilistic Turing machine(PTM)[1]. A PTM is a Turing machine with a random source, and it is a fundamental model for probabilistic computation.

2.1 Time-Bounded Classes

There are several probabilistic analogues of the class **P**. **ZPP**, **RP** or **BPP** are classes of decision problems that can be solved by a polynomial-time PTM with zero-sided error, one-sided error or two-sided error, respectively. The following are more precise definitions for these classes.

[1]See [1, Definition 7.1] for more precise definition.

Definition 1. *A language L is in the class* **ZPP** *if and only if there is an expected polynomial-time PTM M such that*

$$x \in L \quad \Leftrightarrow \quad M(x) = 1.$$

A **ZPP** *algorithm is called a Las Vegas algorithm.*

Definition 2. *A language L is in the class* **RP** *if and only if there is an polynomial-time PTM M such that*

$$\begin{aligned} x \notin L &\quad\Rightarrow\quad M(x) = 0, \textit{ and} \\ x \in L &\quad\Rightarrow\quad \Pr[M(x) = 1] \geq 2/3. \end{aligned}$$

An **RP** *algorithm is called a Monte Carlo algorithm.*

Definition 3. *A language L is in the class* **BPP** *if and only if there is an polynomial-time PTM M such that*

$$\begin{aligned} x \notin L &\quad\Rightarrow\quad \Pr[M(x) = 0] \geq 2/3, \textit{ and} \\ x \in L &\quad\Rightarrow\quad \Pr[M(x) = 1] \geq 2/3. \end{aligned}$$

A **BPP** *algorithm is called an Atlantic City algorithm.*

Obviously, **ZPP** and **BPP** are closed under complementation. **RP**, however, may not be closed under complementation, so there is a dual notion **coRP**. The immediate consequences now follow from the definitions.

Theorem 4. $\mathbf{P} \subseteq \mathbf{ZPP} \subseteq \mathbf{RP} \subseteq \mathbf{BPP}$

Theorem 5. $\mathbf{ZPP} = \mathbf{RP} \cap \mathbf{coRP}$

2.2 Space-Bounded Classes

There are also several probabilistic analogues of the class **L**. **RL** or **BPL** are classes of decision problems that can be solved by a logarithmic-space polynomial-time *PTM* with one-sided error or two-sided error, respectively. Observe that the time is also bounded. The time is automatically bounded for deterministic cases, since a logarithmic-space Turing machine only has a polynomial number configurations. This, however, is not true for probablistic cases. The following are more precise definitions for these classes.

Definition 6. *A language L is in* **RL** *if and only if there is a logarithmic-space polynomial-time PTM M such that*

$$\begin{aligned} x \notin L &\quad\Rightarrow\quad M(x) = 0, \textit{ and} \\ x \in L &\quad\Rightarrow\quad \Pr[M(x) = 1] \geq 2/3. \end{aligned}$$

Definition 7. *A language L is in* **BPL** *if and only if there is a logarithmic-space polynomial-time PTM M such that*

$$\begin{aligned} x \notin L &\quad\Rightarrow\quad \Pr[M(x) = 0] \geq 2/3, \textit{ and} \\ x \in L &\quad\Rightarrow\quad \Pr[M(x) = 1] \geq 2/3. \end{aligned}$$

BPL is obviously closed under complementation. **RL**, however, may not be closed under complementation, so there is a dual notion **coRL**. The immediate consequences now follow from the definitions.

Theorem 8. $\mathbf{L} \subseteq \mathbf{RL} \subseteq \mathbf{BPL}$

2.3 Some Remarks

Although the error bound in the definitions was 1/3, this could be exponentially reduced by repeating the algorithm, and taking a majority vote. More precisely, the error bound becomes $\varepsilon > 0$ if the algorithm is repeated $\lceil \log_3 \varepsilon^{-1} \rceil$-times. Therefore, the error bound could be significantly reduced, so that for all intents and purposes the algorithm may never fail – one would have to wait beyond the life of the universe before the algorithm makes a mistake. This fact implies that randomized algorithms are highly practical.

Moreover, a small error bound of an algorithm implies that most random strings are good. Sometimes, however, it is very difficult to find which random strings are good or not. In other words, derandomization may be difficult. For example, the median finding problem has a fairly simple linear-time randomized algorithm, but all known linear-time deterministic algorithms are relatively complicated and less practical [1, 7.2.1].

The following theorems show that with nonuniformity or nondeterminism, we can efficiently search for the good random strings. The first theorem is due to Leonard Adleman [2] and the second theorem is known as the Siper-Gács theorem [3].

Theorem 9. $\mathbf{BPP} \subset \mathbf{P}_{/\mathbf{poly}}$

Proof. Suppose $L \in \mathbf{BPP}$. Then there is a polynomial-time PTM M for L using an m-bit random string $y \in \{0,1\}^m$ such that

$$\Pr\left[M(x) \neq L(x)\right] \leq 2^{-n-1}$$

for each input size n by the error reduction procedure. Here, it may be assumed that m is polynomial in n, because of the running time of M. For each $x \in \{0,1\}^n$, there are at most 2^{m-n-1} random strings that give a wrong answer. Therefore, at most 2^{m-1} random strings give a wrong answer for some x, meaning there is $y_0 \in \{0,1\}^m$ that always give a correct answer. The circuit C_n for L is now obtained by using y_0, implying $L \in \mathbf{P}_{/\mathbf{poly}}$. □

Theorem 10 (Siper-Gács Theorem). $\mathbf{BPP} \subseteq \mathbf{\Sigma_2^p}$

Sketch of Proof. Suppose $L \in \mathbf{BPP}$. Then there is a polynomial-time TM M for L using an m-bit random string $y \in \{0,1\}^m$ such that

$$\Pr_y\left[M(x,y) \neq L(x)\right] \leq 2^{-n}$$

for each input size n. Let $k = \lceil m/n \rceil + 1$. Then $x \in L$ if and only if

$$\exists u_0, \cdots, u_{k-1} \in \{0,1\}^m \ \forall y \in \{0,1\}^m \ \bigwedge_{i<k} M(x, y \oplus u_i) \text{ accepts}$$

is true. Therefore, $L \in \Sigma_2^p$. See [1, Theorem 7.15] for more detailed proof. □

There is also a known result for **BPL** due to Michael Saks and Shiyu Zhou [4]. The proof was not given in the lecture.

Theorem 11. $\mathbf{BPL} \subseteq \mathbf{DSPACE}(\log^{3/2} n)$

3 Undirected Connectivity

The aim of this section is to prove that the undirected connectivity problem is in the class **RL**. The undirected connectivity problem is a decision problem that decides whether or not two points s and t are in the same connected component of an undirected graph $G = (V, E)$. The problem has a simple linear time algorithm by using DFS or BFS. This algorithm, however, requires a linear space. There is a simple redomized algortihm that only uses logarithmic space.

The idea of the algorithm is fairly simple. It takes a random walk on the graph, and with high probability the walk will reach the terminal vertex if the initial vertex is in the same connected component. More precisely, take a large polynomial l in the input size. Then do a random walk of length l starting from the initial vertex s. The random walk is where at each time, the walk chooses a neighbor of the current vertex uniformly at random to go to. The algorithm accepts the input if the walk reaches the termial vertex t, and rejects if not. Now, we need to analyze this algorithm. This algorithm is due to Romas Aleliunas et al. [5].

Theorem 12. *Let $l = 72n^4 \log n$ where n is the number of vertices. If s and t above are in the same connected component, the algorithm accept with the probability greater than $2/3$. If not, the algorithm always rejects. In other words, the undirected connectivity problem is in* **RL**.

The proof requires some elementary linear algebra. First of all, it may be assumed that the graph is regular with self-loops at each vertex by adding several loops and parallel edges. It may be also assumed that the graph G is conneceted, because the algorithm always returns the correct answer if s and t are in different connected components. Now, let n, m and d be $|V|$, $|E|$ and the degree. Without loss of generality, let the vertex set be of the form of $\{0, 1, \cdots, n-1\}$.

Let v be any distribution over V. Then v could be also considered as a column vector where $\mathrm{v}_i = \Pr[\mathrm{v} = i]$. Let A be the adjacency matrix of G multiplied by a constant scalar $1/d$. Then $A\mathrm{v}$ is a new distribution of v after one random walk step.

The idea of the proof is also simple. It will be shown that after some long random walk, the distribution becomes almost uniform. In other words, if l_0 is large enough then every entry of $A^{l_0}\mathrm{v}$ becomes larger than $1/(2n)$ regardness of the initial distribution v. Then the random walk reaches t within l_0 steps with the probability greater than $1/(2n)$, so reaches t within $3nl_0$ steps with the probability greather than $2/3$. Before we prove the theorem, several lemmata are needed.

Lemma 13. *Let u be the uniform distribution. Then $A\mathrm{u} = \mathrm{u}$, or equivalently, u is an eigenvector of A with the correnponding eigenvalue 1.*

Proof. Since the graph is regular, $(A\mathrm{u})_i$ is just an average of u_j's where j's are neighbors of i. Since the distribution u is uniform, $\mathrm{u}_i = \mathrm{u}_j$, meaning $(A\mathrm{u})_i = \mathrm{u}_i$. Therefore, $A\mathrm{u} = \mathrm{u}$. $\square$

Now, it should be proved that $\|A^l\mathrm{v} - \mathrm{u}\|_2$ is fairly small for large l. This requires several facts about eigenvalues and eigenvectors of A. Note that A has n eigenvalues with corresponding orthogonal eigenvectors, because A is symmetric. Also, since every entry of A is in $[0,1]$ and the sum of entries of each row and column is 1, the eigenvalues should be between -1 and 1. Let the eigenvalues be $\lambda_0, \lambda_1, \cdots, \lambda_{n-1}$ and the corresponding eigenvectors be $\mathrm{x}_0, \mathrm{x}_1, \cdots, \mathrm{x}_{n-1}$. Without loss of generality, it may assumed that $\lambda_{n-1} = 1$, $\mathrm{x}_{n-1} = \mathrm{u}$ and $|\lambda_0| \leq |\lambda_1| \leq \cdots \leq |\lambda_{n-1}| = 1$ by Lemma 13. Also, let $\lambda(G) = |\lambda_{n-2}|$.

Lemma 14. *For every distribution* v *and a natural number* l,

$$\|A^l v - \mathrm{u}\| \leq \lambda(G)^l,$$

where u *is the uniform distribution and the norm here is the Euclidean norm.*

Proof. Let $\mathrm{v} = \sum_{i<n} \alpha_i \mathrm{x}_i$. Since v is a distribution, $\alpha_{n-1} = 1$. Therefore,

$$\begin{aligned}
\|A^l v - \mathrm{u}\|^2 &= \left\| A^l \left(\sum_{i<n} \alpha_i \mathrm{x}_i \right) - \mathrm{u} \right\|^2 \\
&= \left\| \left(\sum_{i<n} \alpha_i \lambda_i^l \mathrm{x}_i \right) - \mathrm{u} \right\|^2 \\
&= \left\| \left(\sum_{i<n-1} \alpha_i \lambda_i^l \mathrm{x}_i \right) + \alpha_{n-1} \lambda_{n-1}^l \mathrm{x}_{n-1} - \mathrm{u} \right\|^2 \\
&= \left\| \sum_{i<n-1} \alpha_i \lambda_i^l \mathrm{x}_i \right\|^2 \\
&= \sum_{i<n-1} |\alpha_i|^2 |\lambda_i|^{2l} \|\mathrm{x}_i\|^2 \\
&\leq \lambda(G)^{2l} \sum_{i<n} |\alpha_i|^2 \|\mathrm{x}_i\|^2 \\
&= \lambda(G)^{2l} \left\| \sum_{i<n} \alpha_i \mathrm{x}_i \right\|^2 \\
&= \lambda(G)^{2l} \|\mathrm{v}\|^2 \\
&\leq \lambda(G)^{2l}.
\end{aligned}$$

This completes the proof. □

Lemma 15. $\lambda(G) \leq 1 - \frac{1}{4dn^2} \leq 1 - \frac{1}{4n^3}$, *where d is the degree of each node in G (including self loops).*

Proof. The proof of this lemma was not given in the lecture. See [1, Lemma 21.4] for proof. (Warning: the book uses a bound on the diameter that doesn't work when self loops are added, hence we get a worse bound on the eigenvalue, but it doesn't end up mattering). □

Proof of Theorem 12. let $l_0 = 8n^3 \log n$. Then

$$\|A^{l_0} v - \mathrm{u}\|_2 \quad < \quad \left(1 - \frac{1}{4n^3}\right)^{8n^3 \log n}$$

$$\leq \frac{1}{n^2}.$$

This implies that for every coordinate i, $|(A^{l_0}v)_i - \mathrm{u}_i| \leq 1/n^2$. Thus the random walk reaches t within l_0 steps with the probability greater than $1/(2n) \leq 1/n - 1/n^2$. Thus the random walk reaches t within $72n^4 \log n \geq 3nl_0$ steps with the probability greather than 2/3. $\square$

References

[1] S. Arora, B. Barak, *Computational Complexity: A Modern Approach*, Cambridge University Press 2009.

[2] Leonard M. Adleman, *Two theorems on random polynomial time*, FOCS 1978:75-83

[3] Michael Sipser, *A Complexity Theoretic Approach to Randomness*, STOC 1983:330-335

[4] Michael E. Saks, Shiyu Zhou, $BP_HSPACE(S) \subseteq DSPACE(S^{3/2})$, J. Comput. Syst. Sci. (JCSS) 58(2):376-403 (1999)

[5] Romas Aleliunas, Richard M. Karp, Richard J. Lipton, László Lovász, Charles Rackoff, *Random walks, universal traversal sequences, and the complexity of maze problems*, FOCS 1979:218-223