18.405J/6.841J Advanced Complexity Theory April 3, 2001

Lecture 14

Lecturer: Daniel A. Spielman

Introduction

We can view random numbers as a resource, so it is important to use as few random numbers as possible. In reality, people don't really use completely random numbers, since completely random numbers are difficult to obtain. One important question is how to get good random bits. The big goal of all of this is determining whether or not BPP or RP is equal to P. In this lecture, we discuss one way of changing a few random bits into many random enough bits in order to reduce the error probability in a BPP algorithm using amplification.

Reusing Random Bits for BPP Algorithms

We want to reduce the error probability in a BPP algorithm from 1/3 to $2^{-k(n)}$. One way is to run the algorithm $O(k(n))$ times and take the majority output. If the algorithm used $f(n)$ bits, we now use $O(f(n)k(n))$ bits. This number of bits, it turns out, can be reduced to $O(k(n)+f(n))$ bits. The method to do this reduction is often called reusing random bits. What we do is we take a certain type of Pseudorandom Number Generator, input $O(k(n)+f(n))$ bits, the PSRG outputs $O(k(n)f(n))$ bits, and then we feed the output bits into the algorithm.

The basic idea is that we first suppose we have a graph on $2^{f(n)}$ vertices, each labeled by a string of length $f(n)$, which has constant degree d, and is a good expander (defined later). Then we do the following:

1. Choose a random vertex, which requires $f(n)$ bits, then output its label.

2. Choose a neighbor of this vertex at random, and output its label. This requires $O(1)$ bits, since we have constant degree.

3. Repeat Step 2 until we have seen k nodes. Then the total number of bits used is $f(n)+O(k(n))$.

The graph is implicitly represented. Some of the ways to construct such a graph look something like: the nodes are vectors in Z_m and the edges are matrices $M_1, ..., M_d$, and then you multiply to get the neighbors.

Definitions

Let $G=(V,E)$ and $N=|V|=2^{f(n)}$. The adjacency matrix A is the NxN matrix where $A_{ij}=1$ if $(i,j)\in E$ and $A_{ij}=0$ otherwise.

Let $\hat{A}=\frac{1}{d}A$ (the sum of the rows and columns in this matrix is 1). Note: $\hat{A}\bar{1}=\bar{1}$, so 1 is an eigenvalue of $\hat{A}$. Let λ be an upper bound on the absolute value of every other eigenvalue. If $\lambda<\frac{1}{32}$, then we'll call G a good expander. If λ is close to 1, then G is a small separator. If λ is small, it means we have to cut lots of edges to get subsets with lots of vertices.

Let B be nodes on which our algorithm gives the wrong answer. Also let B be a NxN matrix such that $B_{ij}=1$ if $i=j\in B$ and $B_{ij}=0$ otherwise. Let $\bar{p}$ be a probability distribution on vertices, so that $p=(p_1,p_2,...,p_N)$, $p_i\geq 0$ and $\sum_i p_i=1$. We also define for all vectors v, $|v|=\sum_i |v_i|$.

Example 1 $|B\bar{p}|$ equals the probability that a node drawn according to $\bar{p}$ lands in B.

Example 2 The vector $\hat{A}\bar{p}$ is the distribution obtained by choosing a node according to $\bar{p}$, and then going to a neighbor at random. For example, if $\bar{p} = (1, 0, ..., 0)^T$, then $\hat{A}\bar{p} = (0, ..., 1/d, 0, ..., 1/d, ..., 0)$. We get a $1/d$ term for each neighbor. If $\bar{p}_0 = (1/N, ..., 1/N)$, then $\hat{A}\bar{p}_0 = \bar{p}_0$ (the same as with $\bar{1}$).

Let $\bar{B}$ denote the good nodes, which are the ones not in B. Let $\bar{B}$ be a matrix in the same way as the matrix B, so we have $\bar{B} + B = I$.

Analysis

The initial probability distribution is $p_o = \frac{1}{N}\bar{1}$. The probability that first node is in B, its neighbor is B, and the next neighbor is in $\bar{B}$ is $|\bar{B}\hat{A}B\hat{A}Bp_0|$.

Lemma 1 *If $B_i \in \{B, \bar{B}\}$ for $i = 1$ to k, then the probability of choosing a random vertex, walking $k-1$ more vertices and landing in set B_i at time i is $|B_k\hat{A}B_{k-1}\hat{A}\cdots\hat{A}B_1\hat{A}p_0|$.*

We now show that the probability of landing in B more than half the time is small.

Lemma 2 ***Key Lemma*** *Assume $\lambda < \frac{1}{32}$. Let the size of the set B be less than $\frac{N}{2^{10}}$. Then for all nonnegative vectors p, $||B\hat{A}p|| \leq \frac{||p||}{16}$.*

From Cauchy-Schwartz: For all p, $|p| \leq N^{\frac{1}{2}}||p||$. This bound is tight for p_0, since $|p_o| = 1$, and $||p_o|| = N^{\frac{-1}{2}}$.

Theorem 3 *The probability that we land in B more than $1/2$ of the time is less than 2^{-k} for any sequence $B_1, ..., B_k$.*

When $B_i \in B$ for more than $1/2$ of the time we get

$$|B_k\hat{A}B_{k-1}\hat{A}\cdots\hat{A}B_1\hat{A}p_0| \leq N^{\frac{1}{2}}||B_k\hat{A}B_{k-1}\hat{A}\cdots\hat{A}B_1\hat{A}p_0||$$

At each step, we either get $B\hat{A}p$ or $\bar{B}\hat{A}p$. In the second case the norm is increasing, so we only worry about the first case, where we use our Key Lemma to get

$$|B_k\hat{A}B_{k-1}\hat{A}\cdots\hat{A}B_1\hat{A}p_0| \leq N^{\frac{1}{2}}||p_0||\left(\frac{1}{16}\right)^{\frac{k}{2}} \leq 4^{-k}$$

There are 2^{k-1} such sequences, by symmetry, so the total probability of getting any such sequence is less than or equal to $2^{k-1}4^{-k} < 2^{-k}$.

Now we go pack and prove the Key Lemma. First we write $\bar{p} = \alpha\bar{1} + \bar{v}$ where $\bar{1}$ and $\bar{v}$ are perpendicular. We then have

$$B\hat{A}\bar{p} = B\hat{A}(\alpha\hat{1}) + B\hat{A}\bar{v}$$
$$||B\hat{A}(\alpha\hat{1})|| = \alpha||B\hat{A}\hat{1}|| = \alpha||B\hat{1}|| \leq \alpha\frac{||\hat{1}||}{32} = \frac{||\alpha\bar{1}||}{32}$$

The last inequality comes from the assumption that B is less than $\frac{1}{2^{10}}$ of the size of N and $32 = \sqrt{2^{10}}$.

To get $||B\hat{A}\bar{v}||$ use $||\hat{A}\bar{v}|| \leq \lambda||\bar{v}|| < \frac{1}{32}||\bar{v}||$. So then we get $||B\hat{A}\bar{v}|| \leq \frac{||\bar{v}||}{32}$.

Putting all of this together we get

$$||B\hat{A}\bar{p}|| \leq ||B\hat{A}(\alpha\hat{1})|| + ||B\hat{A}\bar{v}|| \leq \frac{||\alpha\bar{1}||}{32} + \frac{||\bar{v}||}{32} \leq \frac{||p||}{16}$$