6.841: Advanced Complexity Theory Fall 2012

Problem Set 3 – Communication Complexity and Expander Graphs

Prof. Dana Moshkovitz/TA: Henry Yuen *Due Date: October 25, 2012*

Turn in your solution to each problem on a separate piece of paper. Mark the top of each sheet with the following: (1) your name, (2) the question number, (3) the names of any people you worked with on the problem. We encourage you to spend time on each problem individually before collaborating!

Problem 1 – Some problems in communication complexity

(a) Fix a graph G on n vertices. Suppose that Alice is given a clique C in G, and Bob is given an independent set I in G. Show that the deterministic communication complexity of determining $|C \cap I|$ is $O(\log^2 n)$.

(b) There are two ways you can model randomized communication. In the *private randomness* model, Alice's random bits are hidden from Bob, and vice versa. In the *public randomness* model, Alice and Bob share a random string r that they can both look up. The string r can be arbitrarily long.

Clearly, the public randomness model is more powerful. However, if Alice and Bob are willing to talk just a bit more, they can achieve the same power with only private randomness. Show that any communication protocol Π that uses public randomness and succeeds with probability $1 - \delta$ can be simulated by a protocol Π' that uses only private randomness, succeeds with probability $2/3$, and the $R(\Pi')$ (the communication complexity of Π') is at most $R(\Pi) + O(\log n)$.

Problem 2 – Random walks on expander graphs

You should familiarize yourself with the proof of Theorem 21.12 in your book before working on this problem.

In this problem, you will prove the following: Let G be a d-regular graph on n vertices with normalized second eigenvalue λ. Let $\mathcal{B} \subseteq [n]$ with $|\mathcal{B}| = \beta n$. Let $X_1, \ldots, X_k$ be random variables denoting a $(k-1)$-step random walk on G, where X_1 is a uniformly chosen vertex of G, and X_i is a uniformly chosen neighbor of X_{i-1} in G. Define $B_i = 1$ if $X_i \in \mathcal{B}$, 0 otherwise. Then, for every γ,

$$\Pr\left[\left|\frac{1}{k}\sum_{i=1}^{k} B_i - \beta\right| \geq \lambda + \gamma\right] < 2\exp(-\Omega(\gamma^2 k)).$$

In other words, a random walk on an expander graph will sample a region $\mathcal{B}$ of the graph nearly as well as a truly random selection of $X_1, \ldots, X_k$ would.

(a) Let $B = \sum B_i$. Define the *moment generating function* of B to be $e^{rB} = \prod e^{rB_i}$. Let P be the matrix where $P_{ii} = e^{rb_i}$, where $b_i = 1$ iff vertex $i \in \mathcal{B}$, and 0 everywhere else. Let A be the random walk matrix of G. Show that $\mathbb{E}[e^{rB}] = |(PA)^{k-1}P\mathbf{1}|_1 < \|PA\|_2^k$, where $\mathbf{1}$ denotes the uniform distribution on n elements.

(b) Show that $\|PA\|_2 \leq (1-\lambda)\|PJ\|_2 + \lambda\|PC\|_2$, where J is the $n \times n$ matrix with every entry $1/n$, and C is such that $\|C\|_2 \leq 1$.

(c) Show that $\|PJ\|_2 = 1 + r\beta + O(r^2)$, $\|PC\|_2 = 1 + r + O(r^2)$, and finally $\mathbb{E}[e^{rB}] \leq e^{(\beta+\lambda)rk + O(r^2 k)}$. [Hint: use the Taylor series expansion $e^x = 1 + x + O(x^2)$.]

(d) Via a judicious choice of r, show that $\Pr[B \geq (\beta + \lambda + \gamma)k] \leq e^{-\Omega(\gamma^2 k)}$, and use a symmetry argument to argue $\Pr[B \leq (\beta - \lambda - \gamma)k] \leq e^{-\Omega(\gamma^2 k)}$, to finish the result.

Problem 3 – Derandomizing samplers via expander graphs

In this problem, we're going to explore the random sampling problem and investigate how expanders can reduce the randomness requirements of the problem.

The SAMPLING problem: you're given a function $f : U \to [0, 1]$, where U is some universe of size N and $[0, 1]$ denotes the unit interval on the real line. Your goal is to approximate $\mu(f) = \frac{1}{N} \sum_{u \in U} f(u)$ to within an additive sampling error of ϵ, with probability at least $1 - \delta$ (where the randomness is possibly used for the estimation). We call $1 - \delta$ the *confidence* of your approximation.

(a) Consider the following randomized algorithm SAMPLER1: pick t independent samples $x_1, \ldots, x_t$ from U uniformly at random, and output the average of $f(x_1), \ldots, f(x_t)$. How large do you need to take t in order for the output to be within ϵ of $\mu(f)$, with confidence at least $1 - \delta$? How many bits of randomness did SAMPLER1 require?

(b) Let $S_1(\epsilon, \delta)$ and $R_1(\epsilon, \delta)$ denote the number of samples and random bits required by SAMPLER1, respectively, in order to estimate $\mu(f)$ to within ϵ, with confidence at least $1 - \delta$.

We would like to reduce the randomness requirements of SAMPLER1. Let $r = R_1(\epsilon, 0.01)$, and suppose that we have a expander graph G on 2^r vertices that is d-regular (for constant d) and has normalized second eigenvalue $\lambda < 0.1$.

Using G, design an algorithm SAMPLER2 that draws $O(S_1(\epsilon, 0.01) \log(1/\delta))$ samples from U using $R_1(\epsilon, 0.01) + O(\log(1/\delta))$ random bits, in order to estimate $\mu(f)$ with error ϵ and confidence $1 - \delta$. Compare the randomness used in SAMPLER2 against that of SAMPLER1.

Hint: Use the result from problem 2.

Problem 4 – Limited memory with randomization

A language L is in BPL iff there exists a randomized Turing Machine M that uses $O(\log n)$ space and runs in poly(n) time, where for all x,

$$x \in L \Rightarrow \Pr_r[M(x, r) = 1] \geq 2/3,$$

$$x \notin L \Rightarrow \Pr_r[M(x, r) = 1] \leq 1/3.$$

M has access to its input and randomness via special "input" and "randomness" tapes, which it can query. These tapes don't count towards the space usage of M (the input tape itself exceeds $O(\log n)$ space!). The poly(n) time restriction is important.

(a) Show that $\mathsf{BPL} \subseteq \mathsf{P}$.

A language L is in RL^* iff there exists a randomized Turing Machine M that uses $O(\log n)$ space, where for all x,

$$x \in L \Rightarrow \Pr_r[M(x, r) = 1] \geq 2/3,$$

$$x \notin L \Rightarrow \Pr_r[M(x, r) = 1] = 0.$$

Note two differences between RL^* and BPL – RL^* isn't restricted to running in poly(n) time, and it only has *one-sided error*.

(b) The poly(n)-time restriction on BPL (potentially) makes a big difference: show that $\mathsf{RL}^* = \mathsf{NL}$ (you can find a precise definition of NL in Chapter 4 of the textbook). We currently do not know whether $\mathsf{RL} = \mathsf{NL}$.