Computational Complexity Theory, Fall 2010 November 18

Lecture 20: Error reduction and derandomization

Lecturer: Kristoffer Arnsfelt Hansen Scribe: Camilla Lodahl Gravgaard

Recap from last time

Definition 1 (Expander graphs)**.** G is (n, d, λ) expander if G is a d-regular (multi-)graph with n vertices and $\lambda(G) \leq \lambda$, where $\lambda(G) = \lambda(A) = \max \|Ax\|_2$ where A is the normalized adjacency matrix of G and $u = (\frac{1}{n}, ..., \frac{1}{n})^T$ is the uniform distribution.

Lemma 2 (Expander mixing Lemma)**.** *Let $G = (V, E)$ be (n, d, λ) expander $S, T \subseteq V$. Then*

$$\left| |E(S,T)| - \frac{d}{n} |S| \, |T| \right| \leq \lambda d \sqrt{|S| \, |T|}$$

Deterministic error reduction

Let A be an algorithm that solves a decision problem with error less the $\frac{1}{3}$, by using r random bits. Let G be strongly explicit $(2^r, d, \lambda)$ expander. Vertices in G are identified with strings in $\{0,1\}^r$, and let t be a parameter.

Define an algorithm A' by:

- Choose a random vertex u in G using r random bits.
- Run through all vertices in V of distance t to u. For each of these d^t vertices v, run A with v instead of the random bits (viewing v as a string in $\{0,1\}^r$).
- Return the majority vote of the d^t times we run A.

We use no extra random bits and still only use polynomial time, when we have $t = O(\log n)$. We will show that in this case we can decrease the error to $\frac{1}{n^k}$ for every fixed k.

Theorem 3. *A' solves the same decision problem as A but with an error at most $12\lambda^{2t}$.*

Proof. Let $S \subseteq \{0,1\}^r$ be the strings of random bits where A makes an error, and let $T \subseteq \{0,1\}^r$ be strings of random bits where A' makes an error. Observe first that by assumption we have that $|S| \leq \frac{2^r}{3}$ for A having a probability error $\leq \frac{1}{3}$.

We will now look at G^t instead of G, where G^t is the graph where neighbors have a path of length t between them in G, and we have $\lambda(G^t) = (\lambda(G))^t \leq \lambda^t$.

Since all vertices in T have at least half of all there edges in S (because of the majority vote) we have $|E(S,T)| \geq \frac{d^t}{2}|T|$.

Now we plug these into the statement of the Expander mixing lemma.

For the left side,

$$|E(S,T)| - \frac{d}{2^r}|S||T| \geq \frac{d^t}{2}|T| - \frac{d}{2^r} \cdot \frac{2^r}{3}|T| = \frac{1}{6}d^t|T|$$

And now the right side,

$$\lambda^t \cdot d^t \sqrt{|S||T|} \leq \lambda^t \cdot d^t \sqrt{\frac{2^r}{3}|T|}$$

Now setting left side $\leq$ right side we have

$$\frac{1}{6}d^t|T| \leq \lambda^t \cdot d^t \sqrt{\frac{2^r}{3}|T|} \Rightarrow$$
$$\frac{1}{36}d^{2t}|T|^2 \leq \lambda^{2t} \cdot d^{2t}\frac{2^r}{3}|T| \Rightarrow$$
$$\frac{|T|}{2^r} \leq 12\lambda^{2t}$$

□

If A is poly-time and k is any constant then we can pick $t = \Theta(\log n)$, implying A' poly-time and has error prob $\leq \frac{1}{n^k}$

Randomness efficient error reduction

Let B be an algorithm that solves a decision problem with error ϵ less than $\frac{1}{200}$, by using r random bits (note that we could use the previous deterministic error reduction to ensure this).

Let G be strongly explicit $(2^r, d, \lambda)$ expander, with $\lambda \leq \frac{1}{\sqrt{200}}$. Vertices in G are identified with strings in $\{0,1\}^r$. Let t be a parameter, and define an algorithm $B^{'}$ by:

- Choose a random vertex u in G using r random bits
- Take random-walk in G of length t, by using $O(t)$ extra random bits. Let $u_1, \ldots, u_t$ be vertices the random walk visits.
- Run B for t trials, where each u_i is used once instead of random bits. Return the majority answer.

Theorem 4. *$B^{'}$ solves the same decision problem as B but with an error at most $(\frac{4}{10})^{t/2}$*

Definition 5 (projection)**.** Let $S \subseteq \{0,1\}^r$ be random strings where B gives an error. Define $P : \mathbb{R}^{2^r} \to \mathbb{R}^{2^r}$ as the coordinate projection onto S. We index coordinates by $y \in \{0,1\}^r$. Then P is given by, $P(x)_y = x_y$ if $y \in S$, and 0 otherwise.

Let A be the normalized adjacency matrix for G.

Lemma 6. *For all $x \in \mathbb{R}^{2^r}$ we have $\|PAx\|_2 \leq \sqrt{\epsilon + \lambda^2} \cdot \|x\|_2$.*

Proof. Write $x = x^{\|} + x^{\perp}$ where $x^{\|}$ is parallel with u and $x^{\perp}$ is orthogonal with u. We now use the triangle inequality on the left side and we get:

$$\|PAx\|_2 \leq \|PAx^{\|} + PAx^{\perp}\|_2 \leq \|PAx^{\|}\|_2 + \|PAx^{\perp}\|_2$$

We know that $x^{\|}$ is an eigenvector for A with eigenvalue 1, so we can write:

$$\|PAx^{\|}\|_2 = \|Px^{\|}\|_2 = \|x^{\|}\|_2 \cdot \|P\frac{x^{\|}}{\|x^{\|}\|_2}\|_2 =$$
$$\|x^{\|}\|_2 \cdot \|P(\frac{1}{2^{r/2}}), ..., \frac{1}{2^{r/2}}\|_2 = \|x^{\|}\|_2 \cdot \sqrt{\epsilon \cdot 2^r(\frac{1}{2^{r/2}})^2} = \sqrt{\epsilon}\|x^{\|}\|_2$$

Now we take a look at the orthogonal part. We simply have $\|PAx^{\perp}\|_2 \leq \|Ax^{\perp}\|_2 \leq \lambda\|x^{\perp}\|_2$.

Finally we combine the two expressions, and complete the proof using the Cauchy-Schwartz inequality and Pythagoras.

$$\|PAx\|_2 \leq \sqrt{\epsilon} \cdot \|x^{\|}\|_2 + \lambda\|x^{\perp}\|_2 = \left\langle \begin{pmatrix} \sqrt{\epsilon} \\ \lambda \end{pmatrix}, \begin{pmatrix} \|x^{\|}\|_2 \\ \|x^{\perp}\|_2 \end{pmatrix} \right\rangle \leq$$
$$\left\| \begin{pmatrix} \sqrt{\epsilon} \\ \lambda \end{pmatrix} \right\|_2 \cdot \left\| \begin{pmatrix} \|x^{\|}\|_2 \\ \|x^{\perp}\|_2 \end{pmatrix} \right\|_2 = \sqrt{\epsilon + \lambda^2} \cdot \sqrt{\|x^{\|}\|_2^2 + \|x^{\perp}\|_2^2} = \sqrt{\epsilon + \lambda^2} \cdot \|x\|_2$$

□

Proof. (of Theorem) First we estimate

$$\Pr[B' \text{gives error}] = \Pr[\leq \frac{t}{2} \text{ of } v_1, ..., v_t \text{ gives error for } B]$$
$$\leq \sum_{W \subseteq \{1,...,t\}, |W| \geq \frac{t}{2}} Pr[\forall i \in W : u_i \text{ gives an error for } B]$$

Consider a fixed W. Then

$$\Pr[\forall i \in W : u_i \text{ gives an error for } B] = \|M_t A M_{t-1} A, ..., M_1 Au\|_1 \ ,$$

where $M_i = P$ if $i \in W$ and $M_i = I$ otherwise.

Now we want to change from 1-norm to 2-norm, this is done by multiplying by $\sqrt{2^r}$ so we get a new expression:

$$\|M_t A M_{t-1} A, ..., M_1 Au\|_1 \leq \sqrt{2^r} \cdot \|M_t A M_{t-1} A, ..., M_1 Au\|_2$$
$$\leq \sqrt{2^r}\left(\sqrt{\epsilon + \lambda^2}\right)^{|W|} \cdot \|u_2\| \leq \sqrt{2^r}\left(\sqrt{\frac{1}{200} + \frac{1}{200}}\right)^{|W|} \cdot (\frac{1}{\sqrt{2^r}}) =$$
$$\left(\sqrt{\frac{1}{100}}\right)^{|W|} = \left(\frac{1}{10}\right)^{|W|} \leq \left(\frac{1}{10}\right)^{t/2}$$

Now we are almost at the finish line, look at:

$$\sum_{W \subseteq \{1,...,t\}, |W| \geq \frac{t}{2}} Pr[\forall i \in W : u_i \text{ gives an error for } B] \leq 2^t \left(\frac{1}{10}\right)^{t/2} = \left(2^2\right)^{t/2} \left(\frac{1}{10}\right)^{t/2} = \left(\frac{4}{10}\right)^{t/2}$$

□

Derandomization

Definition 7. A function $G : \{0,1\}^* \to \{0,1\}^*$ is called a cryptographic pseudorandom generator, if there exists another function $l : \mathbb{N} \to \mathbb{N}$ so $l(m) > m$ and $|G(x)| = l(m)$ (l is called the stretch), and it satisfies the following properties:

- G can be computed in polynomial time
- for every probabilistic polynomial time algorithm D we have:

$$|\Pr[D(U_{l(m)}) = 1] - \Pr[D(G(U_m)) = 1]| \leq \frac{1}{m^k}$$

  for every k and every m large enough.

Definition 8 (One-way function)**.** A cryptographic one-way function is a function $f : \{0,1\}^* \to \{0,1\}^*$ where

- f can be computed in polynomial time
- for every probabilistic polynomial time algorithm A we have:

$$Pr[A(f(x), 1^{|x|}) \in f^{-1}(f(x))] \leq \frac{1}{|x|^k}$$

  for every $k \geq 0$ and every n large enough.

An important result of (theoretical) cryptography is an equivalence between existence of pseudo random generators and one-way functions.

Theorem 9. *A cryptographic PRG exists $\Leftrightarrow$ a cryptographic one-way function exists*

For our purposes of derandomization we will need *nonuniform* versions of the above concepts. Specifically, the mention of probabilistic polynomial time algorithms in both definitions are simply to be replaced with polynomial size circuits. The above theorem also holds for the nonuniform setting, that is nonuniform cryptographics PRG's exists if and only if nonuniform cryptographic one-way functions exists.

If we assume the existence of nonuniform cryptographic PRG's we can nontrivially derandomize (i.e. eliminate randomness) the class BPP. The proof is covered in the next lecture.

Theorem 10. *If nonuniform cryptographic one-way functions exists, then*

$$BPP \subseteq \bigcup_{\epsilon>0} DTIME(2^{n^\epsilon})$$