CS 2429 - Foundations of Communication Complexity

Lecture #4: 6 October 2009

Lecturer: Toniann Pitassi

Scribe Notes by: Yuval Filmus

1 The Discrepancy Method — Cont'd

In the previous lecture we've outlined the discrepancy method, which is a method for getting lower bounds on randomized communication complexity given upper bounds on the discrepancy of the matrix M_f corresponding to the function in question. In this lecture we will present two general methods that can be used to upper bound the discrepancy.

We denote the discrepancy of f (with respect to the uniform distribution) and a rectangle $A \times B$ by $\mathrm{disc}_f(A \times B)$. All our results can be generalized to arbitrary distributions by multiplying each entry of M_f by the probability of the corresponding cell.

Recall that Boolean functions can be considered as taking values in either $\{0,1\}$ or $\{+1,-1\}$. In this section, we will use the ± 1 convention.

We use the notation $\mathbf{1}_A$ for the characteristic vector of A, which contains 1 in positions corresponding to the elements of A.

1.1 The Eigenvalue Method

The eigenvalue method upper bounds the discrepancy using the maximal eigenvalue of M_f.

Lemma 1 (Eigenvalue Bound) *Let f be a symmetric Boolean function, i.e. $f(x,y) = f(y,x)$. Then*

$$\mathrm{disc}_f(A \times B) \leq 2^{-2n} \lambda_{\max} \sqrt{|A| \cdot |B|},$$

where $n = |x| = |y|$ is the input size, and $\lambda_{\max}$ is the largest eigenvalue of the symmetric matrix M_f.

Proof Since M_f is symmetric, its eigenvectors v_i form an orthonormal basis for $\mathbb{R}^n$. Denote by λ_i the eigenvalue corresponding to v_i, so that $M_f v_i = \lambda_i v_i$.

Expand the characteristic vectors of A and B in this basis:

$$\mathbf{1}_A = \sum \alpha_i v_i, \qquad \mathbf{1}_B = \sum \beta_i v_i.$$

Putting these expansions into the definition of discrepancy, we are almost done:

$$\begin{aligned} 2^{2n}\mathrm{disc}_f(A\times B) &= \left|\mathbf{1}_A{}^{\mathrm{T}} M_f \mathbf{1}_B\right| \\ &= \left|\left(\sum \alpha_i v_i\right)^{\mathrm{T}}\left(\sum \beta_i \lambda_i v_i\right)\right| \\ &= \left|\sum \alpha_i\beta_i\lambda_i\right| \le \lambda_{\max}\left|\sum \alpha_i\beta_i\right|. \end{aligned}$$

Note that $\sum \alpha_i^2 = \|\mathbf{1}_A\|^2 = |A|$ and similarly $\sum \beta_i^2 = |B|$. The lemma follows from an application of Cauchy-Schwarz:

$$\begin{aligned} 2^{2n}\mathrm{disc}_f(A\times B) &\le \lambda_{\max}\left|\sum \alpha_i\beta_i\right| \\ &\le \lambda_{\max}\sqrt{\sum \alpha_i^2}\sqrt{\sum \beta_i^2} = \lambda_{\max}\sqrt{|A|\cdot|B|}. \end{aligned}$$

We can prove Lindsey's lemma using this bound. First, let's find the eigenvalues of the Hadamard matrices. Recall these are defined by the following recursive construction:

$$H_0 = [1], \qquad H_{n+1} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}.$$

Lemma 2 *For each n, $H_n^2 = 2^n I_{2^n}$.*

Proof The proof is by induction. Since $H_0 = I_1$, the lemma is correct for $n = 0$.

Given that $H_n^2 = 2^n I$, we can calculate H_{n+1}^2 explicitly:

$$\begin{aligned} H_{n+1}^2 &= \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}^2 \\ &= \begin{bmatrix} H_n^2 + H_n^2 & H_n^2 - H_n^2 \\ H_n^2 - H_n^2 & H_n^2 + H_n^2 \end{bmatrix} = \begin{bmatrix} 2^{n+1} I_{2^n} & 0 \\ 0 & 2^{n+1} I_{2^n} \end{bmatrix} = 2^{n+1} I_{2^{n+1}}. \end{aligned}$$

Corollary 3 *The eigenvalues of H_n are all $\pm 2^{n/2}$.*

Moreover, if $n > 0$, then half of the eigenvalues are $+2^{n/2}$, and half of them $-2^{n/2}$.

Proof If λ is an eigenvalue of H_n then λ^2 is an eigenvalue of $2^n I$, so $\lambda^2 = 2^{n/2}$.

Moreover, if $n > 0$ then $\mathrm{Tr}(H_n) = 0$ by construction, and so exactly half of the eigenvalues are positive, and exactly half are negative.

Lindsy's lemma follows:

Lemma 4 (Lindsey's Lemma) *We have $2^{2n}\mathrm{disc}_{\mathrm{IP}_n}(A\times B) \le \sqrt{2^n|A|\cdot|B|}$.*

Here $\mathrm{IP}_n(x,y) = \sum x_i y_i \pmod 2$.

Proof The matrix corresponding to IP_n is H_n. We have shown that $\lambda_{\max}(H_n) = 2^{n/2}$, and so the lemma follows by the Eigenvalue Bound.

1.2 The BNS Method

The BNS method is another way to bound the discrepancy, and will furnish us with yet another proof of the upper bound on $\mathrm{disc}_{\mathrm{IP}_n}$. Actually, it's a general reformulation of our first proof of Lindsey's lemma (given in the previous lecture). The method first appeared in a paper by Babai, Nisan and Szegedy[1].

The method is given by the following lemma:

Lemma 5 (BNS) *The discrepancy of a function $f : X \times Y \to \mathbb{Z}_2$ can be bounded as follows:*

$$\mathrm{disc}_f(A \times B)^2 \leq \mathop{\mathbb{E}}_{y,y'} \left| \mathop{\mathbb{E}}_{x} M_f(x,y) M_f(x,y') \right|,$$

where x, y, y' are chosen independently and uniformly at random, x from X and y, y' from Y.

Proof Recall the definition of discrepancy.

$$\mathrm{disc}_f(A \times B) = \sum_{x \in A, y \in B} M_f(x,y)/2^{2n}.$$

The discrepancy can be written using expectations as

$$\mathrm{disc}_f(A \times B) = \left| \mathop{\mathbb{E}}_{x,y} \mathbf{1}_A(x) \mathbf{1}_B(y) M_f(x,y) \right|.$$

We can recast the Cauchy-Schwarz inequality in the form $\mathbb{E}[Z]^2 \leq \mathbb{E}[Z^2]$. Retracing our steps in the first proof of Lindsey's lemma, we find that

$$\begin{aligned}
\mathrm{disc}_f(A \times B)^2 &= \left(\mathop{\mathbb{E}}_{x} \mathbf{1}_A(x) \mathop{\mathbb{E}}_{y} \mathbf{1}_B(y) M_f(x,y) \right)^2 \\
&\leq \mathop{\mathbb{E}}_{x} \left(\mathbf{1}_A(x) \mathop{\mathbb{E}}_{y} \mathbf{1}_B(y) M_f(x,y) \right)^2 \\
&\leq \mathop{\mathbb{E}}_{x} \left(\mathop{\mathbb{E}}_{y} \mathbf{1}_B(y) M_f(x,y) \right)^2 \\
&= \mathop{\mathbb{E}}_{x} \left(\mathop{\mathbb{E}}_{y,y'} \mathbf{1}_B(y) \mathbf{1}_B(y') M_f(x,y) M_f(x,y') \right) \\
&= \mathop{\mathbb{E}}_{y,y'} \mathbf{1}_B(y) \mathbf{1}_B(y') \left(\mathop{\mathbb{E}}_{x} M_f(x,y) M_f(x,y') \right) \\
&\leq \mathop{\mathbb{E}}_{y,y'} \left| \mathop{\mathbb{E}}_{x} M_f(x,y) M_f(x,y') \right|.
\end{aligned}$$

The bound we get does not depend on the sizes of A and B, and so it is slightly inferior to bounds which do (like Lindsey's lemma). In practice, the difference is usually insignificant (but is the subject of the final question in the first assignment!).

We illustrate the method by proving yet again the upper bound on the discrepancy of the inner product function:

[1] Szegedy is a Hungarian surname and so the digraph *sz* should be pronounced as an English *s* (IPA [*s*]). When an English *sh* sound (IPA [ʃ]) is intended, the monograph *s* is used, for example Erdős.

Lemma 6 *We have* $\mathrm{disc}_{\mathrm{IP}_n}(A \times B) \leq 2^{-n/2}$.

Proof The matrix corresponding to IP_n is H_n. The rows of H_n are orthogonal and so

$$\mathop{\mathbb{E}}_{x} H_n(x,y)H_n(x,z) = \begin{cases} 0 & \text{if } y \neq z, \\ 1 & \text{if } y = z. \end{cases}$$

Using the BNS bound,

$$\mathrm{disc}_{\mathrm{IP}_n}(A \times B)^2 \leq \mathop{\mathbb{E}}_{y,z} \left| \mathop{\mathbb{E}}_{x} H_n(x,y)H_n(x,z) \right| = \Pr[y = z] = 2^{-n}.$$

2 Degree/Discrepancy Method

The Degree/Discrepancy method, due to Sherstov, is a way to come up with functions having high randomized communication complexity. The basic idea is to start with some other function (the "base" function) which is difficult under some other complexity measure, and to "lift" it to a function which is difficult in the randomized communication complexity model. Sherstov's main contribution is using polynomial complexity measures to quantify the difficulty of the base function.

2.1 Polynomial Complexity Measures

We will consider several different complexity measures for the base function. All of them try to capture the notion of being hard to approximate by a polynomial over the real numbers.

Consider a Boolean function $f(x_1, \ldots, x_q)$. We will assume that the inputs and outputs are the usual 0/1 (rather than ± 1). This function can be represented as a real polynomial by following the following steps:

1. Present f as a logical formula, e.g. conjunctive normal form.
2. Convert the formula to a polynomial using the following rules:

$$\begin{aligned} \neg(x) &= 1 - x, \\ x \wedge y &= xy, \\ x \vee y &= x + y - xy. \end{aligned}$$

3. Use the identity $x^2 = x$ to reduce any repeated variables in the monomials.

The result is some polynomial whose degree is at most q.

This prompts the following definition:

Definition The *degree* (also *polynomial degree*) of a function f, written $\deg(f)$, is the minimal degree of a real polynomial P such that $f(x_1, \ldots, x_q) = P(x_1, \ldots, x_q)$ on all Boolean inputs.

In general, it is difficult to represent functions exactly by polynomials, and so the fact that a function has high polynomial degree isn't strong enough for our purposes. A rather lenient alternative is the following:

Definition The *sign degree* (sometimes *polynomial threshold degree*) of a function f, written sign-deg(f), is the minimal degree of a real polynomial P such that for all Boolean inputs $x_1, \ldots, x_q$:

- If $f(x_1, \ldots, x_q) = 1$ then $P(x_1, \ldots, x_q) > 0$.
- If $f(x_1, \ldots, x_q) = 0$ then $P(x_1, \ldots, x_q) < 0$.

This definition is so permissive that it is hard to prove lower bounds on the sign degree. Here are two examples of functions for which a lower bound is known:

- The parity function on q inputs has the maximal sign degree q.
- The Minsky-Papert "tribes" function $\bigvee_{i=1}^{m} \bigwedge_{j=1}^{4m^2} x_{ij}$ has sign degree $m = \sqrt[3]{q/4}$.

Lower bounding the sign degree can be difficult simply because a function with high polynomial degree can be sign-represented by a very low degree polynomial. An extreme example is the OR function (the logical inclusive or of all inputs). This function is sign-represented by the linear polynomial $\sum x_i - \frac{1}{2}$, but an exact representation necessitates a degree q polynomial. This prompts the need for some sort of an interpolation between these two extreme definitions.

The following definition generalizes both previous ones:

Definition [ϵ-Approximation Degree] Given a real $0 \leq \epsilon \leq \frac{1}{2}$, the *$\epsilon$-degree* (more officially, *ϵ-approximation degree*) of a function f, written ϵ-deg(f), is the minimal degree of a real polynomial P such that for all Boolean inputs,

$$|f(x_1, \ldots, x_q) - P(x_1, \ldots, x_q)| \leq \epsilon.$$

If $\epsilon = 0$ this reduces to the regular degree, while if $\epsilon = \frac{1}{2}$ then this (almost) reduces to the sign degree. Clearly the ϵ-degree is monotone decreasing in ϵ, and so for general $0 < \epsilon < \frac{1}{2}$ we have

$$0 \leq \text{sign-deg}(f) \leq \epsilon\text{-deg}(f) \leq \deg(f) \leq q.$$

As an example, the OR function, whose sign-degree is 1 and whose polynomial degree is q, has ϵ-degree $O(\sqrt{q})$ for $\epsilon = 1/8$.

Nisan and Szegedy related the ϵ-degree to decision tree complexity, defined as follows:

Definition A *decision tree* for a Boolean function is a binary tree whose inner vertices are labelled by input variables, and whose leaves are labelled by 0/1. The computation outlined by the tree proceeds from the root by querying the labelled variable, taking the left branch if the respective variable is 0, the right branch if it is 1. Upon reaching a leaf, its label is output.

The *decision tree complexity* of a function f, written DTC(f), is the depth of the shallowest decision tree which represents it.

Using the method outlined above for converting a formula into a real polynomial, one sees that the decision tree complexity upper bounds the polynomial degree. In particular, ϵ-deg(f) $\leq$ DTC(f). Nisan and Szegedy proved a matching upper bound:

$$\epsilon\text{-deg}(f) \leq \text{DTC}(f) \leq \epsilon\text{-deg}(f)^8.$$

Formulated differently, we have $\log \epsilon\text{-deg}(f) = \Theta(\log \text{DTC}(f))$.

2.2 Looking ahead

In the next class we will show how to turn a function with high polynomial degree complexity into a function with high randomized communication complexity. For now, we outline the procedure for a base function with high sign degree. The first observation is the following duality theorem, which we formulate for functions in the ± 1 universe.

Theorem 7 (Duality of Sign Degree) *The function $f(x_1, \ldots, x_q)$ satisfies* sign-deg$(f) > d$ *if and only if there exists a measure μ on $\mathbb{Z}_2^q$ such that*

$$\mathop{\mathbb{E}}_{x_1,\ldots,x_q \sim \mu} [f(x_1, \ldots, x_q)\chi_S(x_1 \ldots, x_q)] = 0 \text{ for all } |S| \leq d,$$

where χ_S, the Fourier character corresponding to S, is defined by

$$\chi_S(x_1, \ldots, x_q) = \prod_{i \in S} x_i.$$

Since the Fourier characters corresponding to all subsets of S span all functions which depend only on inputs from S, the condition in the theorem can be reformulated as

$$\mathop{\mathbb{E}}_{x_1,\ldots,x_q \sim \mu} [f(x_1, \ldots, x_q)g(x_1 \ldots, x_q)] = 0$$

for every function g depending on at most d coordinates.

We will prove this theorem, which is an easy application of linear programming duality, during the next lecture.

Given this theorem, we will proceed according to the following outline:

1. Start with a base function f with high sign-degree.
2. Lift it to a function $F(X, Y)$ defined as follows. The first player's input X is a vector of length $N = q^{O(1)}$. The second player's input Y is an indexing of all possible ordered choices of q bits out of N. Thus $|Y| = \log N!/(N-q)!$. Each input Y defines some sequence of non-repeating indices $0 \leq i_1, \ldots, i_q < N$. We define $F(X, Y) = f(x_{i_1}, \ldots, x_{i_q})$.
3. Use duality to get a distribution, μ, on the inputs of f under which f has zero correlation to all functions depending on few coordinates.
4. Lift this distribution to a distribution, λ, on $X \times Y$ with similar properties.
5. Use the BNS method to upper bound the discrepancy of F with respect to λ, and deduce a lower bound on the randomized communication complexity.