**Harmonic Analysis of Boolean Functions, and applications in CS**

# Lecture 10

*May 12, 2008*

Lecturer: Guy Kindler    Scribe by: Ori Brostovski    Updated: June 11, 2008

In this lecture we describe a permutation test over the long code (Section 2 and then go on to introduce the subject of hardness of approximation and specifically the hardness of *E3-LIN-2* based on the *Unique games conjecture* (Section 3).

## 1 In the previous lecture

In the last lecture we discussed the coordinate permutation test for odd Hadamard codes. Let us recall it.

### Coordinate permutation test

- Codes: $C_1 = C_2 = \{\chi_S\}_{\substack{S\subseteq[n] \\ |S| \text{ odd}}}$ .
- Constraint family: $R_\tau := \{(\chi_S, \chi_{\tau(S)}) : \chi_S \in C_1\}$, where $\tau$ is a permutation over $[n]$.
- Query: Given codewords $f \in C_1$ and $g \in C_2$, the test checks whether for $x \in_R \{\pm1\}^n$, $y \in_R \{\pm1\}^n$, and $u \in_R \{\pm1\}$ the following equation is satisfied:

$$f(x)f(y) = ug(u\tau(xy)) \ .$$

- Completeness: 1.
- Soundness: $\frac{1}{2} + \delta$ (for the coordinate decoding schemes below).

### Decoding schemes

- $D_1(f)$ - Output $\chi_S$ with probability $\hat{f}(S)^2$ ($\perp$ if $|S|$ is even).
- $D_2(g)$ - If $|T|$ is odd, and $\hat{g}(T) \geq \delta$, output $\chi_T$ with probability $\frac{\hat{g}(T)}{\alpha}$ where $\alpha := \sum_{|T| \text{ odd}} \hat{g}(T)$.

## 2 Testing permutations over the long code

In the coordinate permutation test, the test's constraint family corresponded to a set of permutations over $[n]$. That set of permutations can be considered as a subset of the set of permutations over the odd Hadamard code. It is a strict subset as permutations like $\chi_S \to \chi_{[n]\setminus S}$ are not in it. In this section we consider a test that allows checking all possible permutations over long code words. Note that in the case of the long code, the set of coordinate permutation corresponds to the set of permutations over long code words. Before we elaborate on the long code permutation test, we define the distribution $\mu_\epsilon$:

$$\mu_\epsilon = \begin{cases} +1 & \text{w.p. } 1-\epsilon \\ -1 & \text{w.p. } \epsilon \end{cases} .$$

### Long code permutation test

- Codes: $C_1 = C_2 = \{\chi_i\}_{i \in [n]}$.
- Constraint family: $R_\tau := \{(\chi_i, \chi_{\tau(i)}) : \chi_S \in C_1\}$, where $\tau$ is a permutation over $[n]$.
- Query: Given codewords $f \in C_1$ and $g \in C_2$, the test checks whether for $x \in_R \{\pm 1\}^n$, $y \in_R \{\pm 1\}^n$, $z \in_R \{\pm 1\}$, and $z \sim \mu_\epsilon^{(n)}$ the following equation is satisfied:

  $$f(x)f(y) = ug(u\tau(zxy)) .$$

- Completeness: $1-\epsilon$.
- Soundness: $\frac{1}{2} + \delta$ (for the coordinate decoding schemes below).

### Decoding schemes

- $D_1(f)$ - Pick $S$ with probability $\hat{f}(S)^2$. Pick $i \in_R S$, output $\chi_i$.
- $D_2(g)$ - Define $A$ as:

  $$A := \{T : |T| \leq \log_{1-2\epsilon}(0.5\delta) \wedge \hat{g}(T) \geq \delta \wedge |T| \text{ odd}\} .$$

  If $A$ empty, return $\perp$. Else, define $\alpha$ as:

  $$\alpha := \sum_{T \in A} \hat{g}(T) .$$

  Pick $T \in A$ with probability $\frac{\hat{g}(T)}{\alpha}$. Pick $j \in_R T$ and return $\chi_j$.

**Theorem 1** *The long code permutation test has completeness $1-\epsilon$, and soundness $\frac{1}{2} + \delta$ with $\frac{\delta^2}{2\log_{1-2}\ (0.5\delta)}$-satisfaction-rate.*

**Proof**

**Completeness** To show completeness, we assume that we have some $i \in [n]$ such that $f = \chi_i$ and $g = \chi_{\tau(i)}$. We will show that for every $x$ and $y$, the equation

$$f(x)f(y) = g(\tau(xy)) \tag{1}$$

is satisfied. Start from the left hand side:

(Since $f = \chi_i$ and $g = \chi_{\tau_i}$.)

$$f(x)f(y) = \chi_i(x)\chi_i(y) \ ,$$

(Fourier characters are linear.)

$$f(x)f(y) = \chi_i(xy) \ ,$$

(Since $|\tau(i) = 1|$, $\chi_i(u, \ldots, u) = u$.)

$$f(x)f(y) = u\chi_i(u, \ldots, u)\chi_i(xy) \ ,$$

(If a permutation is applied both to a input and to its set, the character's result stays the same.)

$$f(x)f(y) = u\chi_i(u, \ldots, u)\chi_{\tau(i)}(\tau(xy)) \ .$$

We observe that $g(\tau(zxy)) \neq g(\tau(xy))$ if and only if $z_i = -1$. Thus,

$$1 - \epsilon = \Pr[g(\tau(zxy)) = g(\tau(xy))] \ ,$$

(Using equation (1) and our result thus far.)

$$1 - \epsilon = \Pr[g(\tau(zxy)) = f(x)f(y)] \ .$$

**Soundness** As in the soundness proofs of other tests, we develop an expression for $2\Pr[\text{accept}] - 1$:

$2 \cdot \Pr[\text{accept}] - 1 = \mathbb{E}_{x,y,u,z}[f(x)f(y)ug(u\tau(zxy))] \ ,$

(Using Fourier coefficients.)

$$2 \cdot \Pr[\text{accept}] - 1 = \sum_{R,S,T \subseteq [n]} \hat{f}(R)\hat{f}(S)\hat{f}(T)\mathbb{E}_{x,y,u,z}[\chi_R(x)\chi_S(y)u\chi_T(u, \ldots, u)\chi_T(\tau(z))\chi_T(\tau(x))\chi_T(\tau(y))] \ ,$$

(The expectation of $\chi_R(x)\chi_T(\tau(x))$ is 1 if $T = \tau(R)$, and 0 otherwise.)

$$2 \cdot \Pr[\text{accept}] - 1 = \sum_{S \subseteq [n]} \hat{f}(S)^2\hat{g}(\tau(S))\mathbb{E}_{u,z}[u\chi_{\tau(S)}(u, \ldots, u)\chi_{\tau(S)}(\tau(z))] \ ,$$

(The expectation of $u\chi_{\tau(S)}(u,\ldots,u)$ is 1 if $|S|$ is odd, and 0 otherwise. Expectation of a single $\mu_\epsilon$ variable is $(1-2\epsilon)$ which means that the expectation of $\chi_{\tau(S)}(\tau(z))$ is $(1-2\epsilon)^{|S|}$ due to expectation being multiplicative over independent variables.)

$$2\cdot\Pr[\text{accept}]-1=\sum_{\substack{S\subseteq[n]\\|S|\text{ odd}}}\hat{f}(S)^2\hat{g}(\tau(S))(1-2\epsilon)^{|S|}\ .$$

Since we wish to prove soundness, we assume that $\Pr[\text{accept}]>\frac{1}{2}+\delta$, this allows us to say that:

$$\sum_{\substack{S\subseteq[n]\\|S|\text{ odd}}}\hat{f}(S)^2\hat{g}(\tau(S))(1-2\epsilon)^{|S|}\geq 2\delta\ , \tag{2}$$

(Note that $\sum_S\hat{f}(S)^2=1$, and $(1-2\epsilon)\leq 1$. Hence, the total weight of the sets $S$ for which $\hat{g}(\tau(S))<\delta$ is less than $\delta$. We can use this to bound the sum of all sets $S$ in the above sum for which $\hat{g}(\tau(S))\geq\delta$.)

$$\sum_{\substack{S\subseteq[n]\\|S|\text{ odd}\\\hat{g}(\tau(S))\geq\delta}}\hat{f}(S)^2\hat{g}(\tau(S))(1-2\epsilon)^{|S|}\geq\delta\ , \tag{3}$$

(Note that $\sum_S\hat{f}(S)^2=1$, and for every $S$, $\hat{g}(\tau(S))\leq 1$. Hence, the total weight of the sets $S$ for which $|S|<\log_{1-2\epsilon}(0.5\delta)$ is less than $\delta$. We can use this to bound the sum of all sets $S$ in the above sum for which $|S|<\log_{1-2\epsilon}(0.5\delta)$.)

$$\sum_{\substack{S\subseteq[n]\\|S|\text{ odd}\\\hat{g}(\tau(S))\geq\delta\\|S|<\log_{1-2}\ (0.5\delta)}}\hat{f}(S)^2\hat{g}(\tau(S))\geq 0.5\delta\ . \tag{4}$$

Next, we are going to use this bound to finish our proof. We show that the probability that the decoding schemes will return codewords which satisfy the constraint is greater or equal than $\frac{\delta^2}{2\log_{1-2}\ (0.5\delta)}$:

$$\Pr[D_1(f)R_\tau D_2(g)]\geq\sum_{\substack{S\subseteq[n]\\|S|\text{ odd}\\\hat{g}(\tau(S))\geq\delta\\|S|<\log_{1-2}\ (0.5\delta)}}\underbrace{\frac{1}{\alpha}\hat{f}(S)^2\hat{g}(\tau(S))}_{\text{prob. we selected } S \text{ and } \tau(S)}\cdot\underbrace{\frac{1}{|S|}}_{\text{prob. that } j=\tau(i)}\ ,$$

(Since $|S|<\log_{1-2\epsilon}(0.5\delta)$.)

$$\Pr[D_1(f)R_\tau D_2(g)]\geq\sum_{\substack{S\subseteq[n]\\|S|\text{ odd}\\\hat{g}(\tau(S))\geq\delta\\|S|<\log_{1-2}\ (0.5\delta)}}\frac{1}{\alpha}\hat{f}(S)^2\hat{g}(\tau(S))\frac{1}{\log_{1-2\epsilon}(0.5\delta)}\ ,$$

(Using equation (4) and out result thus far.)

$$\Pr[D_1(f)R_\tau D_2(g)] \geq \frac{0.5\delta}{\alpha \log_{1-2\epsilon}(0.5\delta)} \ ,$$

(As shown in the previous lecture, we have $\alpha \leq \frac{1}{\delta}$.)

$$\Pr[D_1(f)R_\tau D_2(g)] \geq \frac{\delta^2}{2 \log_{1-2\epsilon}(0.5\delta)} \ .$$

■

# 3 Hardness of approximation

A variant of theorem (1) can be used to prove an interesting hardness result. We will first explain terms related to hardness of approximation, and then we will describe the result.

## 3.1 Introduction

We will start with the definition of optimization problems and then move on to approimxation problems.

### Optimization problem

- Let $\mathcal{I}$ be a set of instances.
- For every instance $I \in \mathcal{I}$, there is a set of assignments $\mathcal{A}(I)$.
- For every instance $I$ and an assignment $A \in \mathcal{A}(I)$, there is a value $\mathrm{Val}_I(A)$.
- Given an instance $I$, we wish to find $A$ such that:

$$\underset{I}{\mathrm{Val}}(A) = \max_A\{\underset{I}{\mathrm{Val}}(A)\} \ .$$

### Gap problem

- Let $\mathcal{I}$, $\mathcal{A}$, $I$, $A$ and Val be defined the same as in an optimization problem.
- Let $t_1$ and $t_2$ be two scalars such that $t_1 < t_2$.
- We define a function $g : I \to \{0, 1\}$ as following:

$$g(i) = \begin{cases} 1 & \exists A : \mathrm{Val}_I(A) \geq t_2 \\ 0 & \forall A : \mathrm{Val}_I(A) \leq t_1 \\ \text{undefined behaviour} & \text{otherwise} \end{cases} \ .$$

(In some cases $\geq$ may be replaced with $>$ and $\leq$ may be replaced with $<$).

The term *hardness of approximation* refers to the hardness of solving a given gap problem.

## 3.2 Approximating E3-LIN-2

### E3-LIN-2

- An instance $I$ is a distribution over linear equations.
- An assignment $A$ is assignment to the variables of the linear equation.
- $\mathrm{Val}_I(A)$ is defined as:

$$\underset{I}{\mathrm{Val}}(A) := \Pr_{e \sim I}[A \text{ satisfies } e] \ .$$

(For convenience we can think of an instance as a collection of equations with positive weights whose sum is equal to one. In this case $\mathrm{Val}_I(A)$ is the sum of weights of satisfied equations.)

We are interested in the following result:

**Theorem 2** *It is hard to approximate E3-LIN-2 for a gap of* $(\frac{1}{2} + \delta, 1 - \epsilon)$.

In order to prove this result, we define the *unique games* and *label cover* problems.

### Unique games

- Let $k$ be a parameter.
- An instance $I$ is a graph with a set of vertices $V$, a distribution on edges $E$. For each edge $e$ there is some **permutation** $\tau_e \in S_k$.
- An assignment is defined to be a function $A : V \to [k]$ which assigns each vertex with a number.
- $\mathrm{Val}_I(A)$ is defined as:

$$\underset{I}{\mathrm{Val}}(A) = \Pr_{\substack{e \sim E \\ e=(u,v)}} [A(v) = \tau_e(A(u))] \ .$$

When referring to the unique games problem we use the notation $UG[k]$. It is conjectured that:

**Conjecture 3 (Unique games conjecture - Khot, 2002)** *For every $\epsilon$ and $\delta$, there exists $k$ such that $(1 - \epsilon, \delta)$-gap version of UG(k) is $\mathcal{NP}$-hard.*

### Label cover

- Let $k$ be a parameter.
- An instance $I$ is a graph with a set of vertices $V$, a distribution on edges $E$. For each edge $e$ there is some **function** $\pi_e \in [k]^V$.
- An assignment is defined to be a function $A : V \to [k]$ which assigns each vertex with a number.

- $\mathrm{Val}_I(A)$ is defined as:

$$\underset{G}{\mathrm{Val}}(A) = \Pr_{\substack{e \sim E \\ e=(u,v)}} \left[ A(v) = \tau_e(A(u)) \right] .$$

When referring to the label cover problem we use the notation $LC[k]$. It has been proved that:

**Theorem 4** *For every $\epsilon$ and $\delta$, there exists $k$ such that $(1-\epsilon, \delta)$-gap version of LC(k) is $\mathcal{NP}$-hard.*

Theorem (2) can be proven by showing a reduction from the gap version of label cover to E3-LIN-2, and using a variant of theorem (1). In the next class we will show that if conjecture (3) is correct then so is theorem (2).