CSE 533: The PCP Theorem and Hardness of Approximation **(Autumn 2005)**

Lecture 18: Majority Is Stablest and Unique Games Conjecture

30 November 2005

Lecturer: Ryan O'Donnell *Scribe: Ryan O'Donnell and Paul Pham*

1 Recap

Recall from the last class our 2-query "$\neq$" test for binary Long Codes. The test has a "noise parameter" $-1 < \rho < 0$:

- Pick $x \in \{-1, 1\}^m$ uniformly at random.
- Form each bit in $\mu \in \{-1, 1\}^m$ with the following bias:

$$\mu_i = \begin{cases} -1 & \text{with probability } \frac{1-\rho}{2} \\ 1 & \text{with probability } \frac{1+\rho}{2} \end{cases} \tag{1}$$

- Test $f(x) \neq f(x \cdot \mu)$.

We showed that the success probability of the test was related to the "noise stability" of the function, as follows:

$$\Pr[f \text{ passes}] = \frac{1}{2} - \frac{1}{2} \sum_{S \in [n]} \hat{f}(S)^2 \rho^{|S|} = \frac{1}{2} - \frac{1}{2}\text{Stab}_\rho(f). \tag{2}$$

Intuitively, the stability of a function measures its resistance to change when a small number of bits are flipped. Thus, at least for "odd" functions — f satisfying $f(-x) = -f(x)$ — the more noise stable they are, the more likely they are to pass this $\neq$ test.

2 Probability of various functions passing the test

The table below shows the success probability of our 2-query test on various Boolean functions. Recall that another way to look at this probability is the following: Imagine the points of the discrete cube $\{-1, 1\}^m$ lying in m-dimensional Euclidean space on the unit sphere (after scaling). Imagine connecting pairs of points by edges if their inner product is roughly ρ. Then a Boolean function is a cut in this graph (one side is the preimages of -1, the other is the preimages of 1), and the fraction of edges cut corresponds to the probability our test passes.

Note that one interesting class of functions/cuts are those that arise from halfspaces through the origin. These cuts are called halfspace functions or threshold functions, and can be written as $f(x) = \text{sgn}(\sum_{i=1}^m a_i x_i)$.

	Function f	$\Pr[f$ **passes** $]$
1	$f(x) = x_i$	$\frac{1}{2} - \frac{1}{2}\rho$
2	$f(x) \equiv -1$ or 1	0
3	$f(x) = \mathrm{maj}(x) = \mathrm{sgn}\left(\sum_{i=1}^m x_i\right)$	$\frac{\cos^{-1}\rho}{\pi}$ as $m \to \infty$
4	$f(x) = \mathrm{sgn}\left(\sum_{i=1}^m a_i x_i\right)$	$\approx \frac{\cos^{-1}\rho}{\pi}$ for "almost all" choices of a_i's
5	$f(x) = \mathrm{maj}_3(x) = \mathrm{maj}(x_1, x_2, x_3)$	$\frac{1}{2} - \frac{3}{8}\rho - \frac{1}{8}\rho^3$
6	$f(x) = \mathrm{sgn}(Ax_1 + x_2 + x_3 + \ldots x_m)$	A blends between dictator and majority.

(All the functions mentioned above are halfspace functions, except for the constant functions.)

Function 1 is the "ith dictator function", or "ith long code". It is quite easy to check from Equation (2) that the dictator functions (and their negations) are the functions with the highest probability of passing our test. (Remember that $\rho < 0$, so a function wanting to pass the test should try to get as much of its Fourier mass onto "level 1" as possible.) This is certainly something we desire out of a long code test.

Function 2, the constant functions, caused us trouble in our earlier long code test for MAX-3LIN. But in fact for our $\neq$ test, they pass with probability 0, so we needn't worry about them.

Function 3 is the majority function over all m bits in the input. Last lecture we mentioned that this function — which is not at all like a long code — is in some sense the worst case for our test. As mentioned last time, and as we will later sketch with a geometric argument, as $m \to \infty$, the majority function passes our test with probability approaching $\frac{\cos^{-1}\rho}{\pi}$.

In Function 4 we consider "random" or "typical" halfspace functions. One can imagine that a_i's are chosen independently and randomly from some nice probability distribution, such as Gaussians. As we will see later, such functions also pass the test with probability about $\frac{\cos^{-1}\rho}{\pi}$ (with high probability). Thus the name "Majority Is Stablest" is a bit of a misnomer — any "random" halfspace is essentially equally stable.

Function 5, the majority function on just the first three bits, demonstrates that non-long-codes can still pass with probability significantly higher than that of $\frac{\cos^{-1}\rho}{\pi}$. In particular, the graph of $\frac{1}{2} - \frac{3}{8}\rho - \frac{3}{8}\rho^2$ is strictly between that of $\frac{1}{2} - \frac{1}{2}\rho$ and of $\frac{\cos^{-1}\rho}{\pi}$ for all $-1 < \rho < 0$. This means that a non-long-code can pass our $\neq$ test with probability noticeably higher than that for majority. Still, this function is somewhat long-code-like in that it is strongly influenced by only a small constant number of coordinates. We were able to handle similar situations before in the analysis of Håstad's long code test, where it was shown how to disregard functions with large, low-degree Fourier coefficients.

Function 6 takes a parameter $1 \leq A \leq m+1$; if $A = 1$, this is the majority function, if $A = m+1$ this is the first-bit dictator functions. As A grows, we can imagine a blend from majority to dictator, and we can look at the probability the function passes the test. As A increases, the probability goes up, but so does the extent to which f only depends on the first coordinate. As it happens, the critical regime is around $A = \Theta(\sqrt{n})$. Significantly below this, f's success probability is still around $\frac{\cos^{-1}\rho}{\pi}$; when A reaches this range, the success probability starts to get noticeably larger. However, it's also precisely at this point that the first coordinate starts to have a noticeable amount of "influence" over the function. Note that we can't say that f essentially depends only on

x_1 in this case, but at least x_1 is influential.

2.1 Influences

Let us make this notion of "influence" more precise. Recall the following definition from the homework:

Definition 2.1. *The* influence *of the* i*th coordinate on* $f : \{-1,1\}^m \to \{-1,1\}$ *is*

$$\mathrm{Inf}_i(f) := \Pr_x[f(x) \neq f(x^{\oplus i})],$$

where $x^{\oplus i}$ *denotes the string* x *with the* i*th coordinate flipped.*

From the homework, we saw the following:

Proposition 2.2.

$$\mathrm{Inf}_i(f) = \sum_{S \subseteq [m] : i \in S} \hat{f}(S)^2.$$

Some examples of the influences of some of the functions given above are as follows:

$$\mathrm{Inf}_i(x_j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathrm{Inf}_i(\mathrm{maj}_3) = \begin{cases} \frac{1}{2} & \text{if } i \in \{1,2,3\}, \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathrm{Inf}_i(\mathrm{maj}) = \binom{m-1}{\frac{m-1}{2}} 2^{-m} = \Theta(1/\sqrt{m}) \qquad \text{by Stirling's approximation}$$

3 Majority is Stablest

From the above examples we might conjecture that if a function has all of its influences "small", then the probability with which it passes the test cannot be much more than $\frac{\cos^{-1}\rho}{\pi}$. This conjecture was made in [KKMO04], and then proved in [MOO05]. We will sketch the proof in the case that the function is a halfspace cut.

Theorem 3.1 ("Majority is Stablest" (MIS))**.** *Let* $-1 < \rho < 0$ *and* $\epsilon > 0$*. Then there exists* $\tau > 0$*, depending only on* ρ *and* ϵ*, such that if* $f : \{1,-1\}^m \to \{1,-1\}$ *has* $\mathrm{Inf}_i(f) < \tau$ *for all* $i \in [m]$*, then* $\Pr[f \textit{ passes }] < \frac{\cos^{-1}\rho}{\pi} + \epsilon$*.*

Proof. (Partial sketch.) As mentioned, we will sketch the proof in the case that f is a halfspace function, $f(x) = \mathrm{sgn}(\sum_{i=1}^{m} a_i x_i)$. We may assume by scaling all the a_i's that $\sum_{i=1}^{m} a_i^2 = 1$. Under this assumption, it can be checked that if a_i is "large" then $\mathrm{Inf}_i(f)$ is "large". In particular, it can be shown that our assumption that f has all influences smaller than τ implies that all a_i's are smaller than $O(\tau)$ in absolute value.

Let's now examine the probability f passes our test. In our test, we pick random vector $\vec{x} \in \{-1,1\}^m$ according to the uniform distribution, random vector $\vec{\mu} \in \{-1,1\}^m$ according to the $\frac{1-\rho}{2}$-biased distribution, and then test that $\vec{a} \cdot \vec{x}$ and $\vec{a} \cdot (\vec{x}\vec{\mu})$ have different signs. (Here $\vec{x}\vec{\mu}$ denotes the coordinate-wise product.)

Imagine picking $\vec{\mu}$ first, and imagine fixing it to some "typical value". This typical $\vec{\mu}$ will have about a $\frac{1-\rho}{2}$ fraction of -1's. Write $\vec{b}$ for the vector $\vec{a}\vec{\mu}$; i.e., the coordinate-wise product of $\vec{a}$ and $\vec{\mu}$. Conditioned on this $m\vec{u}$, our test now takes the following form: "Pick $\vec{x} \in \{-1,1\}^m$ at random and check that $\vec{a} \cdot \vec{x}$ and $\vec{b} \cdot \vec{x}$ have opposite signs.

With $\vec{a}$ and $\vec{b}$ fixed, let's consider these random quantities, $\vec{a} \cdot \vec{x}$ and $\vec{b} \cdot \vec{x}$. If $a_1 = 1$ and $a_i = 0$ otherwise, the quantity $\vec{a} \cdot \vec{x}$ will be distributed as a random ± 1, as will $\vec{b} \cdot \vec{x}$. We don't actually have to worry about this case though, since we get to assume all a_i's are small in absolute value. As for another case, imagine $a_i = 1/\sqrt{m}$ for all i. Then $\vec{a} \cdot \vec{x}$ is $1/\sqrt{m}$ times the sum of m independent ± 1 (as is $\vec{b} \cdot \vec{x}$). In this case, the Central Limit Theorem tells us that the distribution of $\vec{a} \cdot \vec{x}$ will be very close to that of a Gaussian random variable.

In fact, the Central Limit Theorem actually ensures that so long as all a_i's are "small", the distribution of $\vec{a} \cdot \vec{x}$ will be "close" to that of a Gaussian. (The same holds for $\vec{b} \cdot \vec{x}$.) Furthermore, notice what would happen if we replaced $\vec{x}$ with $\vec{g}$, a random length-m vector with independent *Gaussians* in its coordinates. Then we get $\vec{a} \cdot \vec{g}$, which also has a Gaussian distribution (since the sum of Gaussians is a Gaussian). To make a long story short, a "2-dimensional" version of the Central Limit Theorem ensures that if all a_i's are small in absolute value, the *joint* distribution of $\vec{a} \cdot \vec{x}$ and $\vec{b} \cdot \vec{x}$ is close to the *joint* distribution of $\vec{a} \cdot \vec{g}$ and $\vec{b} \cdot \vec{g}$.

This means that the probability f passes the test (conditioned on $\vec{\mu}$) is close to

$$\Pr[\vec{a} \cdot \vec{g} \text{ and } \vec{b} \cdot \vec{g} \text{ have opposite signs}]. \tag{3}$$

Now $\vec{a}$ and $\vec{b}$ are just two fixed vectors in n-dimensional space, and $\vec{g}$ is in fact a random vector that is equally likely to point in any direction on the n-dimensional sphere. Thus the probability we are considering in (3) is precisely the probability we needed to analyze in the Goemans-Williamson algorithm from last class. There, we showed that the probability was equal to the angle between $\vec{a}$ and $\vec{b}$, over π. I.e.,

$$\Pr[f \text{ passes}] \approx \frac{\angle(\vec{a}, \vec{b})}{\pi} = \frac{\cos^{-1}(\vec{a} \cdot \vec{b})/m}{\pi} = \frac{\cos^{-1}[(\sum_{i=1}^{m} \mu_i)/m]}{\pi}.$$

But with high probability over the choice of μ, $(\sum_{i=1}^{m} \mu_i)/m \approx \rho$.

Thus we conclude that if f is a halfspace cut and all of its influences are small, then the probability that it passes the test is at most $\frac{\cos^{-1}\rho}{\pi}$ plus something small — in fact, more strongly, as we claimed earlier, it is *equal* to $\frac{\cos^{-1}\rho}{\pi}$ plus or minus something small.

□

3.1 Extensions

For our hardness result for MAX-CUT, we will actually need two slight, technical extensions of the MIS Theorem (proved in [MOO05, KKMO04] respectively).

Theorem 3.2. *The MIS Theorem still holds for functions* $f : \{-1, 1\}^m \to [-1, 1]$*. The interpretation we mean here is that for such functions,* $\mathrm{Inf}_i(f)$ *is* defined *by* $\sum_{S \ni i} \hat{f}(S)^2$ *and the success probability of the test is* defined *by* $\frac{1}{2} - \frac{1}{2} \sum_S \rho^{|S|} \hat{f}(S)^2$.

We need this extension because we will later essentially be applying our test to the *averages* of functions.

The second extension requires a new technical definition:

Definition 3.3 (C-degree influence)**.** *Given* $f : \{-1, 1\}^m \to [-1, 1]$ *and an integer* $C \geq 1$*, the* C-degree influence i on f *is:*

$$\mathrm{Inf}_i^{\leq C}(f) = \sum_{S \ni i, |S| \leq C} \hat{f}(S)^2.$$

The extension is that the generalized MIS Theorem 3.2 still holds even if one only assumes that all of f's *low-degree* influences are small. This extension is the theorem we will use next class in our hardness result for MAX-CUT:

Theorem 3.4 (generalized MIS Theorem)**.** *Let* $-1 < \rho < 0$ *and* $\epsilon > 0$*. Then there exists* $\tau > 0$ *and* $C < \infty$ *such that if* f *has* $\mathrm{Inf}_i^{\leq C} < \tau$ *for all* $i \in [m]$*, then* $\frac{1}{2} - \frac{1}{2} \sum \rho^{|S|} \hat{f}(S)^2 < \frac{\cos^{-1} \rho}{\pi} + \epsilon$.

4 Unique Games Conjecture

Using the hardness of Label-Cover, we have a number of optimal inapproximability results for constraint satisfaction problems on 3+ variables; however, it seems difficult to prove such results for 2-variable constraint satisfaction problems such as MAX-CUT. Recognizing this, in 2002 Khot suggested the "Unique Games Conjecture". To state it, we will need to recall the definition of Unique-Label-Cover:

Definition 4.1 (Unique-Label-Cover)**.** *The definition of the Unique-Label-Cover (ULC) problem is the same as Label-Cover, except with the additional constraint that that the constraints must be* bijections *(permutations) on the label set* Σ*. In other words, for each label to one vertex in an edge, there should be a unique label acceptable for the other vertex (in both directions).*

A distinctive property of ULC(Σ) is that the problem of determining whether all constraints can be satisfied is in P. The simple algorithm is as follows: For each connected component in the graph, pick a vertex. Try all possible labels to the vertex and for each, try to label all other connected vertices consistently. This can certainly be done in $\mathrm{poly}(n, |\Sigma|)$ time. Thus we have the following:

Fact 4.2. *For all* $\epsilon > 0$*, Gap-ULC*$(\Sigma)_{1,\epsilon}$ *is* not *NP-hard for any* $|\Sigma|$.

We can now state the Unique Games Conjecture. (Note: the word "Games" comes from the original phraseology in terms of 2-prover 1-round games.)

Definition 4.3. *The Unique Games Conjecture [Kho02] (UGC): For every $\delta > 0$, there exists a sufficiently large m such that Gap-ULC(Σ)$_{1-\delta,\delta}$ is NP-hard, for $|\Sigma| \geq m$.*

Remark 4.4. *The UGC is now a fairly notorious open problem in theoretical computer science. It has been used to show optimal hardness-of-approximation results for problems such as Vertex-Cover, MAX-CUT, MAX-2LIN(q), and others. However there is no compelling evidence that it is true, nor is there compelling evidence that it is false.*

Remark 4.5. *As mentioned, UGC may well be false; that is, Gap-ULC$(\Sigma)_{1-\delta,\delta}$ could be in* P. *The best known algorithmic results for approximating Unique-Label-Cover are due to Charikar, Makarychev and Makarychev [CMM06]. They show that, given a ULC(Σ) instance which is $(1-\delta)$-satisfiable, one can efficiently find a labeling satisfying a $1/|\Sigma|^{\delta/2+O(\delta^2)}$ fraction of the constraints. The algorithm is a significantly souped-up version of the Goemans-Williamson semi-definite programming algorithm. Any improvement on this guarantee would disprove the Unique Games Conjecture, as the next remark shows.*

Remark 4.6. *A particular subproblem of ULC is MAX-2LIN(q), the problem of simultaneously satisfying 2-variable linear equations mod q. [KKMO04] showed the following: UGC $\Rightarrow$ Gap-2LIN(q) with completeness $1-\delta$ and soundness $1/q^{\delta/2+\Omega(\delta^2)}$ is NP-hard. This justifies the last sentence of the previous remark, and also shows that, qualitatively, $1-\epsilon$ versus ϵ' gap hardness of MAX-2LIN(q) is* equivalent *to the UGC.*

Next time, we will show that UGC and the MIS Theorem imply that Gap-MAX-CUT with completeness $\frac{1}{2} - \frac{1}{2}\rho - \epsilon$ and soundness $\frac{\cos^{-1}\rho}{\pi} + \epsilon$ is NP-hard; i.e., UGC implies the Goemans-Williamson algorithm is optimal.

References

[CMM06] M. Charikar, K. Makarychev, and Y. Makarychev. Near-optimal algorithms for Unique Games. 2006.

[Kho02] S. Khot. On the power of unique 2-prover 1-round games. In *34th Annual Symposium on Theory of Computing*, 2002.

[KKMO04] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? 2004.

[MOO05] E. Mossel, R. O'Donnell, and K. Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. In *46th Annual Symposium on Foundations of Computer Science*, 2005.