18.408 Topics in Theoretical Computer Science Fall 2022
Lectures 22,23

Dor Minzer

Today, we will present a reduction from Unique-games to Max-cut which shows that, assuming the Unique-Games Conjecture, the Goemans-Williamson algorithm achieves essentially the best approximation ratio for Max-cut (among polynomial time algorithms). Towards this end we further discuss the long code framework, and the notions of "influences" and "low-degree influences" from analysis of Boolean functions.

1 The Long-code Framework, Influence Style

1.1 Motivation

Recall that the hardness result we showed for 3Lin began by constructing a local tester for the long code with the properties that: (1) codewords of the long code pass the test with probability close to 1, and (2) any word that passes the test with probability significantly more than $1/2$ can be associated with a few (constantly many) long code codewords.

To be more specific, we managed to design a test that if a function $f\colon \{-1,1\}^n \to \{-1,1\}$ passes with probability $1/2+\delta$ then there is a Fourier character $\alpha \in \mathbb{F}_2^2$ of size $O_{\varepsilon,\delta}(1)$ such that $\left|\widehat{f}(\alpha)\right| \geqslant 2\delta$. Thus, we thought of the support of α as the set of potential dictators (with which we can associate longcode codewords) that f is associated with. This is a rather ad-hoc way of arriving at dictator — is there a more natural and direct notion that captures our association to long code codewords?

It turns out that the answer is yes, and in fact this more general notion is critical to prove many other hardness of approximation results. In particular, it is essential for the reduction we will see today from Unique-games to Max-cut. To present, we will first consider the Max-cut problem and design a local tester for the long code using the Max-cut predicate. Such local testers for the long code often go by the name *dictatorship tests*.

1.2 A Dictatorship Tester for Max-cut

Recall that in the Max-cut problem the input is a graph $G=(V,E)$, and we wish to partition the vertices into two sides so that the number of edges crossing from one side to the other is maximized. Alternatively, we may view this problem as a constraint satisfaction problem, as follows. With each vertex $v\in V$ we associate a variable x_v which is supposed to be label by either 1 or -1, and with each equation $e=(u,v)\in E$ we associate the equation $x_u \neq x_v$. The goal now is to label the variables by labels from $\{-1,1\}$ and satisfy as many of the equations as possible. Thus, we see that the predicate corresponding to Max-cut is $P\colon \{-1,1\}^2 \to \{0,1\}$ defined as $P(a,b)=1_{a\neq b}$.

We now wish to design a dictatorship tester for the long-code using this predicate. In other words, we way to construct a distribution μ over $\{-1,1\}^n \times \{-1,1\}^n$ such that:

1. Long code codewords pass the test with probability close to 1: if $f\colon \{-1,1\}^n \to \{-1,1\}$ is a dictator, that is, $f(x) = x_i$ for some $i \in [n]$, then the probability that $P(f(x), f(y)) = 1$ for $(x,y) \sim \mu$ is large c.

2. Far-from-long code codewords pass the test with noticeably smaller probability: if $f\colon \{-1,1\}^n \to \{-1,1\}$ doesn't look like a dictator at all, then the probability that $P(f(x), f(y)) = 1$ for $(x,y) \sim \mu$ is at most s, where s is much smaller than c.

Indeed, constructing such dictatorship tests is often a key step in proving a hardness of approximation results (not only for Max-cut), but in general converting such tests into a proper hardness of approximation results is a non-trivial tasks by itself.

In this language, the power of the Unique-Games Conjecture is that it allows one to bypass this last hurdle, and indeed if one is willing to assume this conjecture there is almost an immediate translation between dictatorship tests and hardness of approximation results. We will not show this connection in full generality and instead focus on the case of Max-cut.

1.3 Influences

To make the question more precise we must clarify what we mean by "doesn't look like a dictator at all". For this we define the notion of influences of coordinates on a function f that capture how much the value of f depends on the ith coordinate of its input.

Definition 1.1. *Let $f\colon \{-1,1\}^n \to \{-1,1\}$ be a function, and $i \in [n]$ be a coordinate. The influence of i is defined as*

$$I_i[f] = \Pr_{x \sim \{-1,1\}^n}[f(x) \neq f(xe_i)],$$

where xe_i is the point x with the ith coordinate flipped.

Note that if f is a dictatorship, say $f(x) = x_1$, then $I_1[f] = 1$ and $I_i[f] = 0$ for any other i. Thus, we can think of the influence of a coordinate i as measuring "how much f is alike the dictator i". Though this is not completely precise, this turns out to be a good and useful notion to consider for the purposes of PCPs. Let us consider a few examples:

1. **Parity functions.** If $f(x) = x_1 x_2 \cdots x_d$, then the influence of each $i \in [d]$ is 1, and the influence of any other variable is 0.

2. **The Majority function.** Suppose n is odd, and define $f(x) = 1$ if $\sum\limits_{i=1}^{n} x_i > 0$ and $f(x) = -1$ otherwise. What is $I_i[f]$? Well, by symmetry it is clear that all of the influences of f are equal, so we fix $i = 1$. Note that sampling $x \sim \{-1,1\}^n$, the probability that $f(x) \neq f(xe_1)$ is the probability that $\sum\limits_{i=1}^{n} x_i$ changes its sign when we change x_1. Thus, it must be the case that $\sum\limits_{i=2}^{n} x_i = 0$, which happens with probability
$$\frac{\binom{n-1}{(n-1)/2}}{2^{n-1}} \approx \sqrt{\frac{2}{\pi}}\frac{1}{\sqrt{n}}$$

Definition 1.2. *Let $f\colon \{-1,1\}^n \to \{-1,1\}$ be a function, $i \in [n]$ be a coordinate and $\tau > 0$. We say f has τ-small influences if for all $i \in [n]$, $I_i[f] \leqslant \tau$.*

1.4 Constructing the Dictatorship Tester

With the notion of influence in mind, we can now re-phrase the question above more precisely. We wish to construct a distribution μ over $\{-1,1\}^n \times \{-1,1\}^n$ such that:

1. Long code codewords pass the test with probability close to 1: if $f\colon \{-1,1\}^n \to \{-1,1\}$ is a dictatorship
$$\Pr_{(x,y)\sim\mu}[f(x) \neq f(y)] \geqslant c.$$
2. Far-from-long code codewords pass the test with noticeably smaller probability: if $f\colon \{-1,1\}^n \to \{-1,1\}$ has τ-small influences, then
$$\Pr_{(x,y)\sim\mu}[f(x) \neq f(y)] \leqslant s + o(1),$$
where the $o(1)$ goes to 0 as τ goes to 0.
3. c and s are far apart.

A natural idea is to take the distribution μ to be the uniform distribution over (x,y) such that $y = -x$, and for this distribution it is clear that one gets that $c = 1$. However, this distribution fails the second property, as any odd function pases this test with probability 1; for example, majority. How can we change this distribution so as to penalize majority (yet keep the performance of dictatorship functions relatively untouched)?

Recall that in the 3-Lin lecture, we wanted to distinguish between low-weight Hadamard codewords and high-weight Hadamard codewords, and for that we applied the noise test. We noticed that long-code codewords only get slightly penalized, whereas high weight Hadamard codewords get heavily penalized. Why shouldn't we try such idea?

More precisely, consider the distribution μ over (x,y) where we pick $x \sim \{-1,1\}^n$ uniformly, set $z = -x$ and then sample y to be a noisy version of z. That is, for each $i \in [n]$ independently set $y_i = z_i$ with probability $1-\varepsilon$, and otherwise sample y_i uniformly from $\{-1,1\}$. In other words, we flip all of the coordinates of x (so that checking "equality" turns into checking "inequality"), and then apply noise.

What can we say about this test, then? If f is a dictatorship, say $f(x) = x_1$, then $f(x) = x_1$ and $f(y) = y_1$, so the test fails only if we resampled the first coordinate and got a different value than the original one, which happens with probability $\varepsilon \cdot \frac{1}{2}$. Hence, we get that $c = 1 - \varepsilon/2$.

What about functions f that have τ-small influences? Intuitively, such functions must depend on many coordinates, so we expect that a slight noise will have several "chances" to change the value of f at a point x. Namely we expect that $f(x) \neq f(y)$ with probability noticeably bigger than ε. Indeed, this is correct and is the content of the "Majority is Stablest" theorem:

Theorem 1.3 (Majority is Stablest)**.** *For all $\varepsilon > 0$ and $\eta > 0$, there is $\tau > 0$ such that the following holds. Suppose that $f\colon \{-1,1\}^n \to \{-1,1\}$ is a function such that $\mathbb{E}[f] = 0$ and $\max_i I_i[f] \leqslant \tau$. Then*
$$\Pr_{(x,y)\sim\mu}[f(x) \neq f(y)] \leqslant 1 - \frac{1}{\pi}\mathsf{Arccos}(1-\varepsilon) + \eta.$$

The proof of this result goes beyond the scope of this course, and we will use it in a black-box way. Recalling that $\mathsf{Arccos}(1-\varepsilon) = \sqrt{2\varepsilon} + O(\varepsilon)$, we get that $s = 1 - \frac{\sqrt{2}}{\pi}\sqrt{\varepsilon} + O(\varepsilon)$, hence we get a gap between c and s in the potential dictatorship test above.

We summarize the properties of the dictatorship test μ:

1. If $f\colon \{-1,1\}^n \to \{-1,1\}$ is a dictatorship, then

$$\Pr_{(x,y)\sim\mu}[f(x) \neq f(y)] \geqslant 1 - \varepsilon/2$$

for large c.

2. If $f\colon \{-1,1\}^n \to \{-1,1\}$ has τ-small influences, for sufficiently small $\tau \leqslant \tau_0(\varepsilon, \eta)$, then

$$\Pr_{(x,y)\sim\mu}[f(x) \neq f(y)] \leqslant \frac{1}{\pi}\mathsf{Arccos}(\varepsilon - 1) + \eta.$$

We will convert this dictatorship test into a hardness of approximation result for Max-cut, and get that approximating Max-cut within any factor larger than $\alpha = \max_{\varepsilon>0} \frac{1-\mathsf{Arccos}(1-\varepsilon)/\pi}{1-\varepsilon/2}$ is NP-hard (assuming the Unique Games Conjecture). Note that α is the approximation ratio that the Goemans-Williamson algorithm achieves, hence we would get that the Goemans-Williamson approximation algorithm for Max-cut is tight!

1.5 A Majority is Stablest Result for Bounded Functions

To use the above ideas in a reduction and to carry out the analysis, it will be necessary for us to consider an arithmetic expression that measure the size of the cut defined by f, that is, $\Pr_{(x,y)\sim\mu}[f(x) \neq f(y)]$. We will also need to generalize this quantity as well as the Majority is Stablest theorem for functions that get values in $[-1,1]$ (as opposed to only $\{-1,1\}$).

Note that for $\{-1,1\}$-valued functions, we have that $f(x)f(y) = -1$ if $f(x) \neq f(y)$ and otherwise it is 1, hence we can write that

$$\Pr_{(x,y)\sim\mu}[f(x) \neq f(y)] = \frac{1}{2}\mathop{\mathbb{E}}_{(x,y)\sim\mu}[1 - f(x)f(y)].$$

The expression on the right hand side makes sense for general functions, and we will use it as our generalization.

Definition 1.4. *Let $\rho \in [0,1]$, and let $x \in \{-1,1\}^n$. The distribution over ρ-correlated points with x, denoted by $\mathrm{T}_\rho x$, is defined by the following randomized process: for each $i \in [n]$ independently, set $y_i = x_i$ with probability ρ, and otherwise sample y_i uniformly from $\{-1,1\}$.*

For $\rho \in [-1,0]$ and $x \in \{-1,1\}^n$, the distribution over ρ-correlated points with x, denoted by $\mathrm{T}_\rho x$, is $-\mathrm{T}_{-\rho}x$. In other words, we sample $y \sim \mathrm{T}_{-\rho}x$ and output $-y$.

With this terminology in mind, we define the stability of f:

Definition 1.5. *Let $\rho \in [-1,1]$ and $f\colon \{-1,1\}^n \to [-1,1]$ be a function. We define*

$$\mathsf{Stab}_\rho(f) = \mathop{\mathbb{E}}_{\substack{x\sim\{-1,1\}^n \\ y\sim\mathrm{T}_\rho x}}[f(x)f(y)].$$

Thus, for Boolean-valued f we have that $\Pr_{(x,y)\sim\mu}[f(x) \neq f(y)] = \frac{1}{2} - \frac{1}{2}\mathsf{Stab}_{-1+\varepsilon}(f)$. With the notion of stability in mind we can state the majority is stablest theorem for bounded functions, but for technical reasons we shall need to replace the notion of influences with the notion of low-degree influences.

1.5.1 Fourier Coefficients, Influences and Low-degree Influences

Recall the discrete Fourier transform of $f\colon \{-1,1\}^n \to \{-1,1\}$ is given as

$$f(x) = \sum_{\alpha\in\mathbb{F}_2^n} \widehat{f}(\alpha)\chi_\alpha(x).$$

The influences of a function f can be related to its Fourier transform as follows:

Claim 1.6. *For $f\colon \{-1,1\}^n \to \{-1,1\}$ and $i \in [n]$, we have*

$$I_i[f] = \sum_{\alpha:\alpha_i=1} \widehat{f}(\alpha)^2.$$

Proof. Note that $f(x) \neq f(xe_i)$ if $\left(\frac{f(x)-f(xe_i)}{2}\right)^2 = 1$, and otherwise $\left(\frac{f(x)-f(xe_i)}{2}\right)^2 = 0$. Hence

$$I_i[f] = \mathop{\mathbb{E}}_x\left[\left(\frac{f(x)-f(xe_i)}{2}\right)^2\right].$$

Consider the function $g(x) = \frac{f(x)-f(xe_i)}{2}$; we will use Parseval's equality to evaluate the last expectation, and for that we compute the Fourier coefficients of g. Expanding the Fourier expansion of f, we have that

$$\begin{aligned} g(x) = \frac{1}{2}\left(\sum_\alpha \widehat{f}(\alpha)\chi_\alpha(x) - \sum_\alpha \widehat{f}(\alpha)\chi_\alpha(xe_i)\right) &= \frac{1}{2}\left(\sum_\alpha \widehat{f}(\alpha)\chi_\alpha(x)(1-\chi_\alpha(e_i))\right) \\ &= \sum_{\alpha:\alpha_i=1} \widehat{f}(\alpha)\chi_\alpha(x), \end{aligned}$$

and the claim follows from Parseval. □

We remark that quantities such as $f(x) - f(xe_i)$ are often thought of as the derivative of f in direction i, and so they make sense for general functions (as opposed to only Boolean valued functions). This can be used to generalize the notion of influence of variables to general functions as norms of this derivative.

In addition, due to the formula above one can ask how much of the influence of f comes from the "low-degree" part of f and how much of it comes from the "low-degree" part, and with respect to it we define low-degree influences:

Definition 1.7. *Let $d \in \mathbb{N}$, $f\colon \{-1,1\}^n \to \mathbb{R}$ and $i \in [n]$. The degree d influence of f is defined as*

$$I_i^{\leqslant d}[f] = \sum_{\substack{\alpha\in\mathbb{F}_2^n:|\alpha|\leqslant d\\ \alpha_i=1}} \widehat{f}(\alpha)^2.$$

We end this section with a simple property of low-degree influences (which is the primary reason we use it instead of influences), stating that there can not be too many variables with large low-degree influence.

Lemma 1.8. *Let $f\colon \{-1,1\}^n \to \mathbb{R}$, $d \in \mathbb{N}$. Then $\sum\limits_{i=1}^{n} I_i^{\leqslant d}[f] \leqslant d\|f\|_2^2$. Consequently, if $f\colon \{-1,1\}^n \to [-1,1]$, then for all $\tau > 0$ the number of coordinates $i \in [n]$ for which $I_i^{\leqslant d}[f] \geqslant \tau$, is at most $\frac{d}{\tau}$.*

Proof. By definition,

$$\sum_{i=1}^{n} I_i^{\leqslant d}[f] = \sum_{i=1}^{n} \sum_{\substack{\alpha : |\alpha| \leqslant d \\ \alpha_i = 1}} \widehat{f}(\alpha)^2 = \sum_{\alpha : |\alpha| \leqslant d} \sum_{i=1}^{n} 1_{\alpha_i = 1} \widehat{f}(\alpha)^2 = \sum_{\alpha : |\alpha| \leqslant d} |\alpha| \, \widehat{f}(\alpha)^2,$$

which is at most $d \sum_{\alpha : |\alpha| \leqslant d} \widehat{f}(\alpha)^2 \leqslant d \sum_{\alpha} \widehat{f}(\alpha)^2 \leqslant d \|f\|_2^2$. □

1.5.2 Majority is Stablest for Bounded Functions

We are now ready to state the Majority is Stablest theorem for bounded functions, and we state it separately for positive ρ's and negative ρ's. For $\rho > 0$, we have:

Theorem 1.9 (Majority is Stablest)**.** *Let* $\rho \in [0, 1]$ *and fix* $\eta > 0$. *Then there are* $d \in \mathbb{N}$ *and* $\tau > 0$ *such that if* $f \colon \{-1, 1\}^n \to [-1, 1]$ *has* $\mathbb{E}[f] = 0$ *and* $\max_i I_i^{\leqslant d}[f] \leqslant \tau$*, then*

$$\mathsf{Stab}_\rho(f) \leqslant 1 - \frac{2}{\pi} \mathsf{Arccos}(\rho) + \eta.$$

The stability of majority. We note that the reason for the name of this theorem is that the stability of the majority function is the right hand side. Indeed, taking $h \colon \{-1, 1\}^n \to \{-1, 1\}$ to be the majority function, that is, $h(x) = 1$ if $|\{ i \mid x_i = 1\}| \geqslant 1$ and otherwise $h(x) = -1$, one has that $\mathsf{Stab}_\rho(h) = \frac{1}{\pi}\mathsf{Arccos}(\rho) + o(1)$: to see that, note that we may define, for each $v \in \{+1, -1\}^n$ the function $h_v \colon \{-1, 1\}^n \to \{-1, 1\}$ which is 1 if $\langle v, x \rangle > 0$ and -1 otherwise. Thus, the majority function is h_v for $v = \vec{1}$, and by symmetry it follows that the stability of all h_v's are the same. Hence,

$$\begin{aligned} \mathsf{Stab}_\rho(\mathsf{Majority}) = \mathop{\mathbb{E}}_{v} \left[\mathsf{Stab}_\rho(h_v) \right] &= \mathop{\mathbb{E}}_{v} \left[1 - 2 \mathop{\Pr}_{(x,y)\ \rho\text{-correlated}} [h_v(x) \neq h_v(y)] \right] \\ &= 1 - 2 \mathop{\mathbb{E}}_{(x,y)} \left[\mathop{\mathbb{E}}_{v} \left[1_{\mathsf{sign}(\langle v,x \rangle) \neq \mathsf{sign}(\langle v,y \rangle)} \right] \right]. \end{aligned}$$

Fixing x and y, we have that $\mathbb{E}_v \left[1_{\mathsf{sign}(\langle v,x \rangle) \neq \mathsf{sign}(\langle v,y \rangle)} \right] \approx \frac{1}{\pi} \theta(x, y) + o(1)$. This is because v can be thought of as a random vector on the unit sphere, and it produces different signs with x and y if and only if they lie in different sides of the hyperplane it is normal to. Since v is a random vector, the hyperplane it is normal to is also random, hence the probability it passes between x and y is proportional to the angle between them. Also, we have that $\theta(x, y) = \mathsf{Arccos}\left(\frac{\langle x,y \rangle}{\|x\|_2 \|y\|_2} \right)$, and $\langle x, y \rangle = (\rho + o(1))n$ with high probability and so $\theta(x, y) \approx \mathsf{Arccos}(\rho)$, so we get $\mathsf{Stab}_\rho(\mathsf{Majority}) \approx 1 - \frac{2}{\pi}\mathsf{Arccos}(\rho)$.

Hence, the above theorem says that the stability of majority is the largest possible (up to $o(1)$) within the class of functions with small influences.

For $\rho \leqslant 0$, we have the following result.

Theorem 1.10 (Majority is Stablest)**.** *Let* $\rho \in [-1, 0]$ *and fix* $\eta > 0$. *Then there are* $d \in \mathbb{N}$ *and* $\tau > 0$ *such that if* $f \colon \{-1, 1\}^n \to [-1, 1]$ *has* $\mathbb{E}[f] = 0$ *and* $\max_i I_i^{\leqslant d}[f] \leqslant \tau$*, then*

$$\mathsf{Stab}_\rho(f) \geqslant \frac{2}{\pi} \mathsf{Arccos}(-\rho) - 1 - \eta.$$

2 A Reduction from Unique-games to Max-cut

2.1 The Starting Point of the Reduction

We first recall the Unique-games problem and the Unique-Games Conjecture, which is the problem we reduce from and the hardness assumption we require to carry out the proof.

Definition 2.1. *An instance of Unique-Games is an instance of Label-cover $\Psi = (G = (L \cup R, E), \Sigma_L, \Sigma_R, \Phi = \{\Phi_e\}_{e \in E})$ wherein $|\Sigma_L| = |\Sigma_R|$ and furthermore each constraint Φ_e is a permutation. That is, for each $e \in E$ there is a 1-to-1 map $\phi_e \colon \Sigma_L \to \Sigma_R$ such that*

$$\Phi_e = \{ (\sigma, \phi_e(\sigma)) \mid \sigma \in \Sigma_L \} .$$

Recall the Unique-games conjecture, which asserts that given a Unique-games instance it is NP-hard to distinguish between the case it is highly satisfiable and the case only a small fraction of the constraints can be satisfied.

Conjecture 2.2. *For all $\varepsilon, \delta > 0$ there is $k \in \mathbb{N}$ such that gap-UniqueGames$[1 - \varepsilon, \delta]$ is NP-hard on instances with alphabet size at most k.*

We will further assume that the Unique-games instances we are dealing with are over regular graphs; this can be added as an assumption if you'd like, however this can also be arranged by standard techniques.

2.2 The Reduction

We are now ready to present the reduction. Let $\rho = 1 - \varepsilon$.

Starting with a Unique-games instance $\Psi = (G = (L \cup R, E), \Sigma_L, \Sigma_R, \Phi)$, we wish to construct a Max-Cut instance with the properties described above. The idea will be to introduce, for each vertex $u \in L$ a separate hybercube $\{-1, 1\}^{\Sigma_L}$, and using a cut in that hypercube to encode the label that u is supposed to get in Ψ. More specifically, we will want to associate with each label σ of u which is supposed to have high value; this will be the dictatorship cut, i.e. the cut defined by $f_u(x) = x_\sigma$. Once we do that, we will be able to argue that if Ψ has a good assignment, then the graph we produce G will have a large cut corresponding to the dictatorship functions in each hypercube.

To ensure soundness, we must take care of two potential issues:

1. Penalizing cuts that are defined by functions that do not "resemble" any dictatorship. We have already dealt with this issue the last section, wherein we argued that in that case the cut size would be at most $1 - \frac{1}{\pi}\mathsf{Arccos}(\rho) + o(1)$ if f does not have any coordinate with significant low-degree influence.
2. Penalizing violating the constraints of Ψ. Namely, suppose we have two vertices $u \in L$, $v \in R$ that have an edge between them, and they have been assigned by dictatorship functions $f_u(x) = x_{\sigma_u}$, $f_v(x) = y_{\sigma_v}$, but σ_v, σ_u do not satisfy the constraint between u and v in Ψ. In that case, we would want to penalize this cut, as it does not correspond to a good assignment in Ψ. To deal with this issue, our edges will not really be inside the hypercube of each vertex v, but rather *across hypercubes*. For that, it is important to note that there is a natural bijection between the hypercube of v and the hypercube of u respecting the constraint between them, which is simply $x \to y$ where $y_i = x_{\phi_{u,v}(i)}$.

This almost finishes the informal overview of the reduction, except that if we were to execute the plan as is, we would get a bipartite graph (the sides being the hypercubes of V and the hypercubes of U), and to remedy that we only leave one of these sides alive, and take two steps in the graph of Ψ instead of one.

We now proceed to the formal construction of the reduction. Given $\Psi = (G = (L \cup R, E), \Sigma_L, \Sigma_R, \Phi)$, we construct a weighted max-cut instance $G = (V', E', w)$ as follows.

- The vertices: For each $u \in L$ we construct a cube over Σ_L, $\{u\} \times \{-1,1\}^{\Sigma}$, which we refer to as the long-code of u. A ± 1 assignment to these vertices should be thought as a potential encoding of one of the labels in Σ_L for u.
- The edges are weighted according to the following randomized process. Sample $v \in R$ and $u, u' \in L$ two neighbours of v independently. Let x be a uniformly chosen vector from $\{-1,1\}^{\Sigma_R}$, and sample $y \sim T_{-\rho} x$. Consider the points

  $$z = \phi_{v,u}(x), \qquad z' = \phi_{v,u'}(y), \qquad \text{where } \phi_{v,u}(y)_\sigma = y_{\phi_{(v,u)}(\sigma)} \; \forall \sigma \in \Sigma_L.$$

  The edge output by the process is (z, z').

We prove the following lemma, encapsulating the analysis of the reduction.

Lemma 2.3. *For all $\rho \in (0,1)$, $\delta > 0$ there is $\eta > 0$ such that:*

1. *Completeness: if Ψ is at least $1 - \eta$ satisfiable, then there is a cut in G of weight at least $\frac{1}{2}(1+\rho) - \delta$.*
2. *Soundness: if Ψ is at most η satisfiable, then G has no cut whose weight exceeds $1 - \frac{1}{\pi}\mathsf{Arccos}(\rho) + \delta$.*

2.3 Analysis of the reduction

We now analyze the construction. First, we show the completeness of the construction, asserting that if Ψ is highly satisfiable, then there exists a large cut on the graph we have constructed.

2.4 Completeness

Suppose there are labelings $A_L \colon L \to \Sigma_L$ and $A_R \colon R \to \Sigma_R$ satisfying at least $1 - \eta$ fraction of the edges. We assign ± 1 values to the cube of u according to the dictatorship assignment of $A(u)$. Namely, we define the cut in the graph G by

$$f(u, x) = x_{A_L(u)} \text{ for } (u, x) \in V \times \{-1,1\}^{\Sigma_L}.$$

We analyze the weight of the cut defined by f. Looking at the process describing the weights of the edges in G', Since the graph of Ψ is regular, the marginal distribution of each one of the edges $(v,u), (v,u')$ is uniform; therefore the probability both are satisfied by A_L and A_R is at last $1 - 2\eta$. Sample x, y as in the process, and look at $\phi_{(v,u)}(x), \phi_{(v,u')}(y)$. Note that $y_{A_R(v)} \neq x_{A_R(v)}$ with probability $\frac{1}{2} + \frac{1}{2}\rho$, and if that happens, since both edges (v,u) and (v,u') are satisfied, we get that

$$f(u,z) = z_{A_L(u)} = z_{\phi_{v,u}(A_L(u))} = x_{A_R(v)} \neq y_{A_R(v)} = z'_{\phi_{v,u'}(A_L(u'))} = f(u', z').$$

We conclude that the weight of edges crossing the cut is at least $\frac{1}{2} + \frac{1}{2}\rho - 2\eta$.

2.5 Soundness

In this part, we show that if the UG instance Ψ had no good satisfying assignments then the graph G does not have a large cut. We prove it in a counter-positive way: assuming we have a large cut in the graph, we will construct a good assignment for Ψ.

Let $f\colon L \times \{-1,1\}^{\Sigma_L} \to \{-1,1\}$ be a function corresponding to a large cut, that is a cut of size at least $\frac{1}{\pi}\mathsf{Arccos}(\rho) + \delta$. The fractional size of the cut is exactly

$$\Pr_{\substack{v,u,u' \\ x,y,z,z'}}\big[f(u',z') \neq f(u,z)\big].$$

Let ν be a vector from $\{-1,1\}^{\sigma}$ such each coordinate is -1 with probability $\frac{1}{2}(1-\rho)$. Then the previous probability is the same as

$$\Pr_{\substack{v,u,u' \\ x,\nu}}\big[f(u,\phi_{(v,u)}x) \neq f(u',\nu \cdot \phi_{(v,u')}x)\big].$$

Define for $u \in U$, $v \in V$

$$g_v(x) = \mathop{\mathbb{E}}_{u:(u,v)\in E}\big[f(u,\phi_{(v,u)}x)\big], \qquad\qquad g_u(x) = f(u,x).$$

Intuitively, v asks his neighbours what side it should be on, and takes the average of the suggestions. Then

$$\begin{aligned}
\Pr_{\substack{u,v,v' \\ x,\nu}}\big[f(u,\phi_{(v,u)}x) \neq f(u',\nu \cdot \phi_{(v,u')}x)\big] &= \frac{1}{2}\left(1 - \mathop{\mathbb{E}}_{\substack{v,u,u' \\ x,\nu}}\big[f(u,\phi_{(v,u)}x)f(u',\nu \cdot \phi_{(v,u')}x)\big]\right) \\
&= \frac{1}{2}\left(1 - \mathop{\mathbb{E}}_{\substack{v \\ x,\nu}}\left[\mathop{\mathbb{E}}_{u}\big[f(u,\phi_{(v,u)}x)\big]\mathop{\mathbb{E}}_{u'}\big[f(u',\phi_{(v,u')}(\nu \cdot x))\big]\right]\right) \\
&= \frac{1}{2}(1 - \mathop{\mathbb{E}}_{\substack{v \\ x,\nu}}[g_v(x)g_v(\nu \cdot x)]) \\
&= \frac{1}{2}(1 - \mathop{\mathbb{E}}_{v}[\mathsf{Stab}_{-\rho}[g_v]]).
\end{aligned}$$

We conclude that since the fractional size of the cut is at least $1 - \frac{1}{\pi}\mathsf{Arccos}(\rho) + \delta$, it holds that

$$\mathop{\mathbb{E}}_{v}[\mathsf{Stab}_{-\rho}[g_v]] < \frac{2}{\pi}\mathsf{Arccos}(\rho) - 1 - 2\delta.$$

We say v is good if $\mathsf{Stab}_{-\rho}[g_v] \leqslant \frac{1}{\pi}\mathsf{Arccos}(\rho) - \delta$. Note that by an averaging argument, it follows that at least δ fraction of the $v \in L$ are good, and we denote the set of these by L_{good}. We fix d,τ corresponding to ρ,δ as in Theorem 1.10 and apply it to get that there is i such that $I_i^{\leqslant d}[g_v] \geqslant \delta$ for each $v \in L_{\mathsf{good}}$. Define

$$\mathsf{List}_{\xi}(v) = \left\{ i \mid I_i^{\leqslant d}[g_v] \geqslant \xi \right\}.$$

Since the sum of the d degree influence is at most d, $|\mathsf{List}(v)| \leqslant d/\xi$; the important point is that this quantity only depends on ρ,ε (and not on $|\Sigma_L|$). We finish by showing that if v is good and $i \in \mathsf{List}_{\tau}(v)$, then a non-negligible fraction of his neighbours u have $\phi_{(v,u)}(i) \in \mathsf{List}_{\tau/2}(u)$. To see that we first prove a simple connection between the Fourier coefficients of g_v's and g_u's:

Claim 2.4. *For all $\alpha \in \mathbb{F}_2^{\Sigma_R}$ we have $\widehat{g_v}(\alpha) = \mathbb{E}_{u:(u,v)\in E}\left[\widehat{g_u}(\phi_{(v,u)}\alpha)\right]$.*

Proof.

$$\begin{aligned}\widehat{g_v}(\alpha) = \mathop{\mathbb{E}}_{x}\left[g_v(x)\chi_\alpha(x)\right] = \mathop{\mathbb{E}}_{x}\left[\mathop{\mathbb{E}}_{u}\left[g_u(\phi_{(u,v)}x)\right]\chi_\alpha(x)\right] &= \mathop{\mathbb{E}}_{u}\left[\mathop{\mathbb{E}}_{y}\left[g_u(y)\chi_\alpha(\phi^{-1}_{(u,v)}y)\right]\right]\\ &= \mathop{\mathbb{E}}_{u}\left[\mathop{\mathbb{E}}_{y}\left[g_u(y)\chi_{\phi_{(u,v)}\alpha}(y)\right]\right]\\ &= \mathop{\mathbb{E}}_{u}\left[\widehat{g_u}(\phi_{(u,v)}\alpha)\right]\end{aligned}$$

The second equality is by the definition of g_v, the third equality is since $\chi_\alpha(\phi x) = \chi_{\phi^{-1}\alpha}(x)$ for a permutation ϕ. □

Lemma 2.5. *Suppose $v \in L_{\mathsf{good}}$, and let $i \in \mathsf{List}_\tau(v)$. Then*

$$\Pr_{u:(u,v)\in E}\left[\phi_{v,u}(i) \in \mathsf{List}_{\tau/2}(u)\right] \geqslant \frac{\tau}{2}.$$

Proof. By definition and Claim 2.4 we get that

$$\tau \leqslant I_i^{\leqslant d}[g_v] = \sum_{\alpha:|\alpha|\leqslant d,\alpha_i=1} \widehat{g_v}^2(\alpha) = \sum_{\alpha:|\alpha|\leqslant d,\alpha_i=1} \mathop{\mathbb{E}}_{u:(u,v)\in E}\left[\widehat{g_u}(\phi_{(v,u)}\alpha)\right]^2,$$

and so by Jensen's inequality we conclude that

$$\begin{aligned}\tau \leqslant \sum_{\alpha:|\alpha|\leqslant d,\alpha_i=1} \mathop{\mathbb{E}}_{u:(u,v)\in E}\left[\widehat{g_u}(\phi_{(v,u)}\alpha)^2\right] &= \mathop{\mathbb{E}}_{u:(u,v)\in E}\left[\sum_{\alpha:|\alpha|\leqslant d,\alpha_i=1} \widehat{g_u}(\phi_{(v,u)}\alpha)^2\right]\\ &= \mathop{\mathbb{E}}_{u:(u,v)\in E}\left[\sum_{\beta:|\alpha|\leqslant d,\beta_{\phi_{v,u}(i)}=1} \widehat{g_u}(\beta)^2\right]\\ &= \mathop{\mathbb{E}}_{u:(u,v)\in E}\left[I^{\leqslant d}_{\phi_{v,u}(i)}[g_u]\right].\end{aligned}$$

As $I^{\leqslant d}_{\phi_{v,u}(i)}[g_u] \leqslant 1$ always, it follows that with probability at least $\tau/2$ over the choice of u we have that $I^{\leqslant d}_{\phi_{v,u}(i)}[g_u] \geqslant \tau/2$. □

Randomized assignment to the Unique-Games instance

Now we finish the proof. For each $v \in L_{\mathsf{good}}$ assign a label $i \in \mathsf{List}_\tau(v)$ randomly, and for each $u \in R$ assign a label from $\mathsf{List}_{\tau/2}(u)$ randomly. We now lower the probability a randomly chosen edge from Ψ is satisfied.

Choose (u, v) randomly. With probability at least δ, the vertex v is good and we choose some label $i \in \mathsf{List}_\tau(v)$ for it. We condition on v and i. By Lemma 2.5, it follows that with probability at least $\tau/2$ over the choice of u we have that the label $\phi_{v,u}(i)$ is in $\mathsf{List}_{\tau/2}(u)$, and as that list contains at most $d/(\tau/2)$ elements it follows that we have assigned the label $j = \phi_{v,u}(i)$ to u with probability at least $\frac{\tau/2}{d}$.

We conclude that, in expectation over the choice of the assignment, the probability that a random edge is satisfied is at least

$$\delta \cdot \frac{\tau}{2} \cdot \frac{\tau/2}{d} = \delta'(\delta, \rho) > 0$$

hence this is smaller than the soundness of the original Unique-games instance provided that $\eta < \delta'$. Hence, we conclude that if the original Unique-games instance was at most η satisfiable for sufficiently small η, then the graph G' we produce has no cut of size $\frac{1}{2}\left(1 - \frac{1}{\pi}\mathsf{Arccos}(\rho)\right) + \delta$. This completes the proof of Lemma 2.3.

Remark 2.6. *We stress here an important point, which is that the performance of the randomized strategy we found for the Unique-games may depend on various parameters we have used in the reduction (such as the noise rate ρ). Most importantly though, it does not depend on the alphabet size of the Unique-games instance, so if we take the soundness of the Unique-games instance to be small enough (which naturally would mean its alphabet size is also large) we would reach a contradiction. This is very typical to hardness of approximation result that use list-decoding arguments such as above (namely, an argument that is able to produce a short list of candidate labels for a vertex and then chooses one randomly), highlighting the importance of "dimension-free" result in Fourier analysis (such as the Majority is Stablest theorem).*

18.408 Topics in Theoretical Computer Science: Probabilistically Checkable Proofs
Fall 2022