CS 880: Advanced Complexity Theory	3/7/2008
Lecture 18: Inapproximability of MAX-3-SAT	
Instructor: Dieter van Melkebeek	Scribe: Jeff Kinne

In this lecture we prove a tight inapproximability result for MAX-3-SAT. We first prove a tight inapproximability result for MAX-3-LIN and derive the result for MAX-3-SAT with a gap preserving reduction. This is a general approach that can be used to prove inapproximability results for many NP-complete problems. At the end of this lecture we also state inapproximability results for MAX-CUT and MIN-VC that can be derived using a similar strategy. However these results are not tight. We develop tight inapproximability results for the latter two problems in future lectures, under the unique games conjecture (a stronger assumption than P $\neq$ NP which we will also state in future lectures).

We began developing the inapproximability result for MAX-3-LIN in the previous lecture. The main task today is to complete this construction.

1 Basic Setup

The starting point for inapproximability results is usually a strong statement of the PCP Theorem that is a statement of the existence of a gap inducing reduction from 3-SAT to constraint graph games (CGG). An inapproximability result for the problem of interest is then derived with a gap-preserving reduction from CGG.

An instance of the constraint graph game is a bipartite graph with edge constraints. The game consists of assigning values to vertices to maximize the number of edge constraints that are satisfied. We use L to denote the vertices on the left side of the graph, and each vertex $u \in L$ can be given a value in $[\ell]$ for some constant $\ell > 0$; R denotes the vertices of the right side of the graph, and each vertex $v \in R$ can be given a value in $[r]$ for some $r > 0$. Each edge $e = (u, v) \in E$ has a constraint function c_e, a Boolean function indicating if the values given to u and v satisfy the constraint. The value of the game, denoted $\nu(G)$, is the maximum fraction of edge constraints that can be simultaneously satisfied.

A game G is of *function type* if c_e can be described as a function π_e from $[\ell]$ to $[r]$, meaning that any value to u determines the value that must be given to v to satisfy c_e. The game is of *unique* or *permutation* type if for each edge e, π_e is a permutation. Unique games are the subject of the unique games conjecture, and will be treated in future lectures. Today, we focus on games of function type, which are the subject of the strong PCP Theorem.

Theorem 1 (Strong PCP Theorem)**.** *For any constant $\gamma > 0$, there is a polynomial time transformation from 3-CNF formulas ϕ to constraint graph games G of function type having possible vertex values from $[\ell]$ and $[r]$ with $\ell, r \leq \frac{1}{\mathrm{poly}(\gamma)}$ such that*

$$\phi \in SAT \Rightarrow \nu(G) = 1,$$

$$\phi \notin SAT \Rightarrow \nu(G) \leq \gamma.$$

Notice that this automatically implies an inapproximability result (assuming P $\neq$ NP) for CGG: any approximation of $\nu(G)$ to within any non-trivial constant factor can be used to distinguish between the two cases. Our goal today is a gap preserving reduction from CGG to 3-LIN – that is a method to approximate $\nu(G)$ when given an algorithm that approximates MAX-3-LIN.

2 Gap Preserving Transformation to 3-LIN: First Attempt

Recall that an instance of 3-LIN is a system of linear equations each over three variables. As we use $\{-1, 1\}$ values, we are dealing with equations of the form $x \cdot y \cdot z = \pm 1$. For MAX-3-LIN, the goal is to satisfy as many of the equations simultaneously as possible. At least half of the equations can be simultaneously satisfied for any set of equations. Our gap preserving reduction from CGG to 3-LIN consists of a reduction from CGG to 3-LIN such that if we can satisfy the resulting instance of 3-LIN on much more than half of its equations, we can satisfy a fraction larger than γ of the CGG constraints. Together with the strong PCP Theorem, this will imply that if P $\neq$ NP, then MAX-3-LIN cannot be approximated to within a constant factor better than half.

Inspiration for the reduction comes from the fact that we have seen a procedure that uses tests which are precisely linear tests over three variables. Namely, the linearity test consists of such tests, and we also developed a test for the long code (equivalent to testing for dictatorship) by adapting this test. We develop the transformation from CGG to 3-LIN in a similar way – using the long code and linear tests, and making modifications as they become necessary.

We will take a CGG G and map it to a system of linear equations S. For each vertex $u \in L$, we include a function $f_u : \{-1, 1\}^\ell \to \{-1, 1\}$ with the intention that the characteristic sequence of f_u is the long code of a value to assign to vertex u. We similarly include a function $g_v : \{-1, 1\}^r \to \{-1, 1\}$ for each vertex $v \in R$. The goal is to create variables for the values of f_u and g_v and create a system of linear equations such that satisfying the linear equations is equivalent to both: 1) enforcing that f_u and g_v are long codes of some values i and j, and 2) checking that assigning values i and j to vertices u and v satisfies the constraint c_e if $e = (u, v)$ is present in the game G. Since we can assume that G is of function type, the former is equivalent to checking that $\pi_e(i) = j$. We can get this to look like a test for linearity by observing that c_e is met if and only if

$$(\forall y \in \{-1, 1\}^r) f_v(y \circ \pi_e) = g_v(y), \tag{1}$$

where $y \circ \pi_e$ is an ℓ bit string that is the result of applying π_e and then y, so $(y \circ \pi_e)_i = y_{\pi_e(i)}$. Notice that if f_u and g_v correspond to long codes of i and j as intended, this simplifies to checking that for all y, $y_{\pi_e(i)} = y_j$, thus providing a sanity check that checking (1) is the right thing to do.

If we pick y at random, the right-hand side of (1) amounts to sampling g_v at a random point. We would like the left-hand side to similarly correspond to sampling f_u at a random point, but $(y \circ \pi_e)$ in general will not be uniformly distributed. We can employ the self-correcting trick that we used when designing the test for dictatorship to make the sampling point uniform. If we do this, we get the following test.

- **Variables**: $f_u(x)$ for all $u \in L$ and $x \in \{-1, 1\}^\ell$; $g_v(y)$ for all $v \in R$ and $y \in \{-1, 1\}^\ell$.
- **Test**: $T^{f,g}$

(1) Pick $e = (u, v) \in E$ at random

(2) Pick $x \in \{-1,1\}^\ell$ and $y \in \{-1,1\}^r$ at random
(3) Accept if and only if $f_u((y \circ \pi_e) \cdot x) \cdot f_u(x) = g_v(y)$

We can convert these tests into a system of linear equations by iterating over all choices of e, x, and y and including the corresponding equation for each. As the size of x and y are constants, this is a polynomial time procedure, and we get the types of equations we desired – namely linear equations of three variables each.

2.1 Completeness

The test has perfect completeness: if $\nu(G) = 1$, then setting all variables according to the genuine long codes of a labeling that realizes $\nu(G)$, satisfies all of the equations, i.e., f and g can be set so that $T^{f,g}$ accepts with probability 1. However, it is also true that the equations can all be satisfied no matter what the value of $\nu(G)$ – by setting all variables to 1. This indicates already that we will have to modify the test in some way.

2.2 Soundness

We want to show that if we have an advantage in satisfying the equations of our tests, then we can use this advantage to satisfy many constraints in the game G. As we can always satisfy at least half of the equations, we assume f and g such that

$$\Pr_{e,x,y}[T^{f,g} \text{ accepts}] \geq \frac{1}{2} + \delta.$$

By Markov's inequality, we have that for a fraction at least $\delta/2$ of $e = (u,v) \in E$,

$$\Pr_{x,y}[T^{f,g} \text{ accepts}] \geq \frac{1}{2} + \frac{\delta}{2},$$

which is equivalent to

$$\mathrm{E}_{x,y}[f_u((y \circ \pi_e) \cdot x) \cdot f_u(x) \cdot g_v(y)] \geq \delta. \tag{2}$$

By Fourier expansion, we have that

$$\mathrm{E}_{x,y}[f_u((y \circ \pi_e) \cdot x) \cdot f_u(x) \cdot g_v(y)] = \mathrm{E}[\sum_{\substack{S,T \subseteq [l] \\ U \subseteq [r]}} \widehat{f_u}(S)\chi_S((y \circ \pi_e) \cdot x)\widehat{f_u}(T)\chi_T(x)\widehat{g_v}(U)\chi_U(y)]. \tag{3}$$

Note that

$$\begin{aligned} \chi_S(x + y \circ \pi_e) &= \chi_S(x)\chi_S(y \circ \pi_e) \\ &= \chi_S(x) \prod_{i \in S} (y \circ \pi_e)_i \\ &= \chi_S(x)\chi_{\pi_e{}'(S)}(y), \end{aligned}$$

where $\pi'_e(S)$ are the values of $[r]$ that are hit an odd number of times by $\pi_e(S)$: $\pi'_e(S) = \{j \in [r] \mid |\pi_e(S)| \text{ is odd}\}$. Standard (by now) harmonic analysis techniques then let us simplify (3) to

$$\mathrm{E}_{x,y}[f_u((y \circ \pi_e) \cdot x) \cdot f_u(x) \cdot g_v(y)] = \sum_{S \subseteq [\ell]} (\hat{f}_u(S))^2 \cdot \hat{g}_v(\pi'_e(S)).$$

By (2) we conclude that for a fraction at least $\frac{\delta}{2}$ of the edges $e = (u, v) \in E$

$$\sum_{S \subseteq [\ell]} (\hat{f}_u(S))^2 \cdot \hat{g}_v(\pi'_e(S)) \geq \delta. \tag{4}$$

We call such e good edges. Our aim is to use (4) to derive values for the vertices of G that will satisfy many constraints. The novel idea that leads to the final test (which is coming shortly) is to view the terms $(\hat{f}_u(S))^2$ and $\hat{g}_v(\pi'_e(S))$ as probability distributions and assigning weights to values based off of these. By Parseval's equality the former is a valid probability distribution, but the latter is not. We use the Cauchy-Schwarz inequality to convert the expression into one where both terms are valid probability distributions

$$\sum_{S \subseteq [\ell]} (\hat{f}_u(S))^2 \cdot \hat{g}_v(\pi'_e(S)) \leq \sqrt{\sum_S (\hat{f}_u(S))^2} \cdot \sqrt{\sum_S (\hat{f}_u(S))^2 \cdot (\hat{g}_v(\pi'_e(S)))^2}.$$

As $\sum_S (\hat{f}_u(S))^2 = 1$, we conclude that for good edges e

$$\sum_S (\hat{f}_u(S))^2 \cdot (\hat{g}_v(\pi'_e(S)))^2 \geq \delta^2. \tag{5}$$

Now both terms are valid probability distributions, one over sets in $[\ell]$ and the other over sets in $[r]$.

Choosing Values

Having derived the inequality with both expressions being valid probability distributions, the following strategy for choosing values for the game G is natural.

- Choose $S \subseteq [\ell]$ with probability $(\hat{f}_u(S))^2$, and choose $T \subseteq [r]$ with probability $(\hat{g}_v(T))^2$.
- Choose i at random from S and j at random from T.
- Output the values i for vertex u and j for vertex v.

The second step is not well-defined if either S or T are empty. This will not be a problem in the final test. For now, we ensure this procedure is always well defined by choosing i or j completely at random if either S or T is empty.

Given this method of choosing values, we can lower bound the probability that these values satisfy the constraint c_e as follows for a fixed good edge $e = (u, v)$.

$$\begin{aligned}
\Pr[c_e \text{ is satisfied}] &= \Pr[\pi_e(i) = j] \\
&\geq \Pr[\pi_e(i) = j \text{ and } T = \pi'_e(S)] \\
&= \textstyle\sum_{S \subseteq [\ell]} \Pr[S] \cdot \Pr[T = \pi'_e(S) | S] \cdot \Pr[\pi_e(i) = j | S \text{ and } T = \pi'_e(S)],
\end{aligned}$$

where the second line follows because adding conditions never increases probability, and the last line is by conditioning on S. We can plug in the values $\Pr[S] = (\hat{f}_u(S))^2$ and $\Pr[T = \pi'_e(S)|S] = (\hat{g}_v(\pi'_e(S)))^2$ for the first two terms. The last term is at least $\frac{1}{|S|}$ by the definition of π'_e provided $\pi'_e(S)$ is non-empty. We conclude that for good edges e

$$\Pr[c_e \text{ is satisfied}] \geq \sum_{S \subseteq [\ell] s.t. \pi'_e(S) \neq \emptyset} (\hat{f}_u(S))^2 \cdot (\hat{g}_v(\pi'_e(S)))^2 \cdot \frac{1}{|S|}. \tag{6}$$

3 Gap Preserving Transformation to 3-LIN

The previous sections set the general framework of our reduction from CGG to 3-LIN. In this section, we finish the construction by making two modifications.

3.1 First Modification

(6) almost gives a good bound on the probability of satisfying a good edge: (5) provides the final link, except that there the $\frac{1}{|S|}$ term is missing. We would like to either get rid of this term in (6) or introduce this term into (5). The latter amounts to introducing a term that gets smaller with large S, and is something we have seen before in designing the test for dictatorship out of the linearity test. There, we used the noise operator T_α to introduce a term of $\alpha^{|S|}$.

Our first modification to the test of the previous section is to use the same trick here. Namely, in the test $T^{f,g}$ we replace the occurrence of $f_u(x)$ by instead applying f_u to z where z is obtained from x by independently flipping each bit with probability ϵ, where $\epsilon = \frac{1-\alpha}{2}$. As when we used this trick in the dictatorship test, this has the effect of replacing one occurrence of $\hat{f}_u(S)$ with $\alpha^{|S|}\hat{f}_u(S)$ in the derivation leading up to (4), and with $\alpha^{2|S|}\hat{f}_u(S)$ in (5).

3.2 Second Modification

We still have one hurdle to overcome: it may be that a large weight in $\hat{g}$ is placed on sets S with $\pi'_e(S) = \emptyset$. We can prevent this from happening by forcing $\hat{g}_v(\emptyset) = 0$, meaning the expected value should be 0. We could ensure this by replacing $g_v(y)$ by $\mathrm{E}_{b\in\{-1,1\}}[b \cdot g_v(b \cdot y)]$. The effect is replacing g by its odd part, so that in particular the Fourier coefficient of $\emptyset$ is 0. By doing this, $\hat{g}_v$ is 0 for each S in the entire derivation leading up to (5).

3.3 Final Test

By incorporating these two modifications into the test originally given, we get the following test.

- **Variables**: $f_u(x)$ for all $u \in L$ and $x \in \{-1,1\}^\ell$; $g_v(y)$ for all $v \in R$ and $y \in \{-1,1\}^\ell$.
- **Test**: $T_\epsilon^{f,g}$

(1) Pick $e = (u, v) \in E$ at random
(2) Pick $x \in \{-1,1\}^\ell$, $y \in \{-1,1\}^r$, and $b \in \{-1,1\}$ at random
(3) Pick $z \sim_\epsilon x$ for $\epsilon = \frac{1-\alpha}{2}$
(4) Accept if and only if $f_u((y \circ \pi_e) \cdot x) \cdot f_u(z) = b \cdot g_v(b \cdot y)$

3.3.1 Completeness

Suppose $\nu(G) = 1$. If we choose f and g as valid long codes as before, the only reason $T_\epsilon^{f,g}$ does not accept is if the bit of x corresponding to the value of u is flipped when creating z. Thus, these f and g satisfy

$$\Pr[T_\epsilon^{f,g} \text{ accepts}] \geq 1 - \epsilon.$$

3.3.2 Soundness

Suppose there are f and g such that $\Pr[T_\epsilon^{f,g} \text{ accepts}] \geq \frac{1}{2} + \delta$. We have observed that in this case at least $\frac{\delta}{2}$ fraction of edges are good. By choosing values for the vertices in the game G as described in the previous section and making use of the second modification to the test, a lower bound on the probability of satisfying a good edge is given by (6) with the exception that the condition $\pi'_e(S) = \emptyset$ never occurs. Due to the first modification, (5) becomes $\sum_S \alpha^{2|S|}(\hat{f}_u(S))^2(\hat{g}_v(\pi'_e(S)))^2 \geq \delta^2$. Provided $\frac{1}{|S|} \geq \alpha^{2|S|}$, we put this all together to conclude that there exist f and g such that the expected value of the game when choosing values as described is at least $\frac{\delta}{2} \cdot \min_S(\frac{1}{|S| \cdot \alpha^{2|S|}}) \cdot \delta^2$. Recall that $\alpha = 1 - 2\epsilon$. Straightforward calculus shows that $\min_S(\frac{1}{|S| \cdot \alpha^{2|S|}}) \geq \Omega(\epsilon)$. We conclude that

$$\nu(G) \geq \beta\epsilon\delta^3,$$

for some constant $\beta > 0$.

If we assume an unsatisfiable formula ϕ in the reduction from 3-SAT to CGG in the strong PCP Theorem, it must be that $\nu(G) < \gamma$, meaning that $\beta\epsilon\delta^3 \leq \gamma$.

3.3.3 Conclusion

For simplicity, we set $\epsilon = \delta$. Then if we start with a 3-SAT formula ϕ which is satisfiable and compose the reduction of the Strong PCP Theorem with the reduction we have just given, then the MAX-SAT of the resulting system of linear equations (implied by the test $T_\epsilon^{f,g}$) is at least $1 - \delta$. If we start with an unsatisfiable formula, our analysis shows that $\delta \leq \sqrt[4]{\gamma/\beta}$ and the MAX-SAT of the system of equations we have generated is at most $\frac{1}{2} + \sqrt[4]{\gamma/\beta}$.

Because we can make both δ and γ constants as small as we like, we conclude the following.

Theorem 2. *If* P $\neq$ NP*, then MAX-3-LIN cannot be approximated to within* $1/2 + \epsilon$ *for any constant* $\epsilon > 0$.

We point out that both of the modifications we made to the original test are necessary. The first modification breaks perfect completeness of the system of equations, which we know is necessary (unless P = NP) because it can be decided in polynomial time whether a 3-LIN instance is fully satisfiable or not. Without the second modification, the linear equations could always be satisfied by setting all variables to 1 (meaning both completeness and soundness would be 1, and we would have no gap).

4 Implications

As with standard reductions between NP problems, once we have a few hardness of approximation results, we can derive others by using appropriate reductions. Among the simplest reductions from MAX-3-LIN is the reduction to MAX-3-SAT. We reduce to 3-SAT by creating four 3-clauses for each linear equation, with one clause each for the four possible assignments to the variables of the left-hand side of the equation that would give an answer different than the bit of the right-hand side of the equation. In the instance of 3-LIN, close to half of the linear equations could not be satisfied. Now at least one in four of the clauses generated by each of these cannot be satisfied, giving the following.

Theorem 3. *If* $P \neq NP$*, then MAX-3-SAT cannot be approximated to within* $7/8 + \epsilon$ *for any constant* $\epsilon > 0$*.*

We point out that this result is tight because there is a 7/8 approximation for MAX-3-SAT. More complicated reductions can be given for the problems MAX-CUT and MIN-VC (among others), giving the following inapproximability results.

Theorem 4. *If* $P \neq NP$*, then MAX-CUT cannot be approximated to within* $16/17+\epsilon$ *and MIN-VC cannot be approximated to within* $10\sqrt{5} - 21 - \epsilon$ *for any constant* $\epsilon > 0$*.*

These results are not tight: currently, the best approximation algorithm for MAX-CUT guarantees a factor of roughly .878, and for MIN-VC the best known achievable factor is 2. (The constants above are $16/17 \approx .941$ and $10\sqrt{5} - 21 \approx 1.36$.)

In the next few lectures, we develop tight results for MAX-CUT and MIN-VC, although under a stronger assumption – the unique games conjecture. For each of these problems, the equations that naturally arise involve two rather than three variables (for MAX-CUT specifying that an edge between two vertices crosses the cut, and for MIN-VC that an edge is covered by one of two vertices). The dictatorship test we used today requires three queries and thus three variables per equation. Next time we will see that we can replace that test with a noise sensitivity test, which only makes two queries and therefore results in two variables per equation.