Georgia Tech Fall'04
CS 8002 : PCPs and Hardness of Approximation

Lecture 7: Hardness of Clique

Lecturer: *Subhash Khot* Scribe: *Vangelis Markakis*

In this lecture we will present hardness results for the MAX-CLIQUE problem:

Definition 1 (Max-clique) *Given an n-vertex graph G, find a maximum size clique in G, i.e., a complete subgraph of maximum size.*

We will first show a constant factor hardness based on the FGLSS reduction [3]. We will then see how to boost the hardness to n^{ϵ}, for some constant ϵ by using the FGLSS reduction, the PCP theorem and random walks on expanders. Finally we will use the PCP construction by Samorodnitsky and Trevisan [7] to obtain a hardness of $n^{1-\epsilon}$ for any $\epsilon > 0$ (this is simpler than Håstad's original PCP [4] for $n^{1-\epsilon}$ hardness for Clique).

1 FGLSS reduction and constant factor hardness for MAX-CLIQUE

Before we state the result that we are going to prove, we need to introduce some notation and terminology. Consider a PCP verifier that asks q queries to the prover. We will first introduce the notion of *free* queries.

Definition 2 *If S is a set of queries asked by a PCP verifier, then $S' \subseteq S$ is a set of free queries if an accepting answer to all queries in $S \backslash S'$ is determined by the answers to the queries in S' and the acceptance predicate.*

For example, consider Hastad's 3-bit PCP, which has a linear acceptance test. Suppose the test is of the form $x + y + z = 0$ in $GF(2)$. Then $\{x, y\}$ is a set of free queries because once we know the answers to x and y, we also know what value z should have for the verifier to accept. We will use f to denote the number of free queries ($f = 2$ in the example). The following is a trivial observation:

Fact 1
The number of accepting answers for a verifier with f free queries is 2^f.

If the acceptance test of the verifier is not linear, then we define f to be equal to the logarithm of the number of accepting answers.

Definition 3 *We will denote by $PCP_{c,s}[r, q, f]$ the class of all languages L that have a PCP verifier that uses r random bits, examines q queries from the proof, the number of free queries is f ($f \leq q$), and the completeness and soundness parameters are c, s respectively ($s < c$).*

DEFINITION 4 *A tuple (S,a) is called an accepting pattern, if for some choice of a random string by the verifier, S is the set of query locations ($|S|=q$), and $a \in \{0,1\}^q$ is an accepting answer.*

Notice that to prove hardness for MAX-CLIQUE, it suffices to prove hardness for the maximum independent set problem, which we will denote by MAX-IS, since approximating MAX-CLIQUE in G is the same problem as approximating MAX-IS in the complement graph of G. This will be done using the following theorem, known as the FGLSS reduction [3].

THEOREM 1
Let $L \in PCP_{c,s}[r,q,f]$. Then there is a polynomial time reduction from L to MAX-IS that maps an instance x to a graph G_x such that:

- $x \in L \Rightarrow \exists$ *I.S.* $I \subseteq G_x, |I| \geq c\ 2^{-f}|G_x|$
- $x \notin L \Rightarrow \forall$ *I.S.* $I \subseteq G_x, |I| \leq s\ 2^{-f}|G_x|$

If L is an **NP**-hard language, then the reduction claimed in the Theorem would imply that Gap-MAX-IS$_{c/s}$ is **NP**-hard. Hence MAX-IS and MAX-CLIQUE would be hard to approximate with a factor better than c/s.
PROOF: Consider a language $L \in PCP_{c,s}[r,q,f]$. We will prove that for every proof Π, there is a corresponding I.S. $I \subseteq G_x$ (and vice versa) such that:

$$\mathbf{Pr}[\Pi \text{ is accepted on input } x] = \frac{|I|}{|G_x|}2^f$$

Given an instance x, we construct a graph G_x as follows: Let $\tau_1, ..., \tau_{2^r}$ be the 2^r possible random strings used by the verifier. The graph G_x has a set of vertices for every such random string. We will denote by T_i the set of vertices corresponding to the string τ_i. Each such set has 2^f vertices. Each vertex from the set T_i corresponds to one of the 2^f possible accepting answers from a proof, given that the verifier used the string τ_i. Hence in total the graph has 2^{r+f} vertices. Moreover each vertex corresponds to an accepting pattern (S,a). Vertices from the same set T_i will have the same set S since the verifier used the same random string to generate the query locations and each a will be one of the 2^f accepting answers, given S. Two vertices $(S,a),(S',a')$ will form an edge in G_x if and only if $S \cap S' \neq \emptyset$ and the answers a and a' assign different values to at least one bit location in $S \cap S'$, i.e., the accepting patterns are conflicting. For example, two vertices from the same set T_i are always conflicting because they have the same set of query locations and there is at least one bit on which they assign different values. Hence:

FACT 2
The vertices (S,a) that belong to the same set T_i, $1 \leq i \leq 2^r$, form a clique.

LEMMA 2
For any I.S. $I \subseteq G_x$, there is a proof Π such that $\mathbf{Pr}[\Pi$ is accepted on input $x] \geq \frac{|I|}{2^r} = \frac{|I|}{|G_x|}2^f$.

PROOF: Pick a maximal independent set I. It can contain at most one vertex from each set T_i. The patterns from each vertex of I are not conflicting, therefore we can construct a proof by patching the patterns together and picking the rest of the bits arbitrarily. This proof is accepted for every random string τ_i for which the correspnding set T_i has a vertex in I. Therefore

$$\mathbf{Pr}[\Pi \text{ is accepted on input } x] \geq \frac{|I|}{2^r}$$

□

In fact, the converse is also true.

LEMMA 3
For any proof Π, *there is an independent set* I *such that* $|I| = 2^r \mathbf{Pr}[\Pi$ *is accepted on* $x]$.

PROOF: Given Π, we form I as follows: for every set T_i, we check if there is a vertex $(S, a) \in T_i$ such that a agrees with Π (clearly there can be at most one vertex from each T_i). If yes, include this vertex in I. It is easy to see then that I is equal to the number of random strings for which the verifier accepts. Thus $|I| = 2^r \mathbf{Pr}[\Pi$ is accepted on $x]$. □

In order to finish the proof of Theorem 1, let $x \in L$. Since $L \in PCP_{c,s}[r, q, f]$, there is a proof π such that $\mathbf{Pr}[\Pi$ is accepted$] \geq c$. But by Lemma 3, there exists an independent set I, such that $|I| \geq c2^{-f}|G_x|$. On the other hand, if $x \notin L$, then for every proof Π, $\mathbf{Pr}[\Pi$ is accepted$] \leq s$. But by Lemma 2 it follows that for every independent set I, $|I| \leq s2^{-f}|G_x|$. □

By the PCP Theorem, $SAT \in PCP_{1,1/2}[O(\log n), O(1), O(1)]$. Therefore this establishes a hardness factor of 2 for MAX-CLIQUE.

COROLLARY 4
There is no $(2 - \epsilon)$*-factor approximation algorithm for MAX-CLIQUE, unless* $\mathbf{P} = \mathbf{NP}$.

PROOF: Immediate from Theorem 1, since the size of the graph produced is $2^{r+f} = n^{O(1)}$. □ In fact if we run the SAT verifier k times independently and accept only if the verifier accepts in all k times, we can see that $SAT \in PCP_{1,1/2^k}[O(k \log n), O(k), O(k)]$. If we use this new verifier in the FGLSS reduction, then as long as k is a constant, the size of the graph produced is still polynomial and is equal to $2^{r+f} = n^{O(k)} = n^{O(1)}$. The hardness factor is then 2^k, for any constant k.

COROLLARY 5
There is no constant factor approximation algorithm for MAX-CLIQUE, unless $\mathbf{P} = \mathbf{NP}$.

2 Hardness of n^ϵ

In the previous section, we could not prove a superconstant hardness result because by running the SAT verifier k times independently, the verifier needs a total of $O(k \log n)$ random bits. Hence as k becomes superconstant, the correspondng size of the graph produced from the FGLSS reduction gets superpolynomial. We will see how to run the verifier k times by using only $O(\log n) + O(k)$ random bits. Although the k random strings used by the verifier will not be independent, we will see that they behave almost as if they were. The crucial theorem that we will use is the following:

Theorem 6

For any r, there exists a bipartite graph $G = (V, W, E)$ such that every vertex in V has degree k, $|W| = 2^r$, $|V| = 2^R$, $R = r + O(k)$ and for any $W' \subseteq W$ with $|W'| = s|W|$ (say $s = 1/2$)

$$\mathbf{Pr}_v[N(v) \subseteq W'] \leq s^{\Omega(k)} \tag{1}$$

Here $N(v)$ denotes the set of neighbors of v.

The proof of Theorem 6 is in [6]. The idea of the proof is that we first construct a constant degree expander on 2^r vertices and then for each $v \in V$, we take a random walk on the expander of length k and include as neighbors of v in G all the vertices visited in this random walk. It is then shown that $R = r + O(k)$ suffices to give the desired graph. For our purposes, we can think of the 2^r vertices of W as all the possible random strings for a SAT verifier that uses r random bits.

Theorem 7

If $SAT \in PCP_{1,s}[r, q, f]$ then $SAT \in PCP_{1,s^{\Omega(k)}}[r + O(k), kq, kf]$.

Proof: Suppose $SAT \in PCP_{1,s}[r, q, f]$ and let X be a verifier for SAT with completeness 1 and soundness s and parameters $[r, q, f]$. Construct the graph $G = (V, W, E)$ guaranteed by Theorem 6. Think of the vertices in W as all the possible 2^r random strings that can be used by X. We construct a new verifier Y which will do the following:

- Pick $v \in V$ at random.
- Let $w_1, ..., w_k$ be its k neighbors.
- Run X on each w_i, $1 \leq i \leq k$, i.e., run X using the random string corresponding to the vertex w_i.
- Accept if X accepts in all k runs.

The new verifier only picks v at random. Hence Y needs $r + O(k)$ random bits. Let ϕ be a satisfiable formula. Then there exists a proof such that X accepts with probability 1 and Y also accepts the same proof with probability 1. Now suppose ϕ is not satisfiable. Since the soundness parameter of X is s, for any proof, there are at most $s2^r = s|W|$ random strings that make the verifier accept. Let W' be the set of such strings with regard to some proof Π. Then the probability that Y accepts Π is the probability that the neighborhood of v falls inside W'. But we know by Equation 1 that this probability is at most $s^{\Omega(k)}$. Therefore $SAT \in PCP_{1,s^{\Omega(k)}}[r + O(k), kq, kf]$. □

Now let us pick $k = \log n$. Since $SAT \in PCP_{1,1/2}[O(\log n), O(1), O(1)]$, we have that $SAT \in PCP_{1,1/2^{\Omega(\log n)}}[O(\log n), O(\log n), O(\log n)]$, where n is the size of the SAT instance. Using Theorem 1, it is hard to approximate MAX-CLIQUE with factor $2^{\Omega(\log n)} = n^{\Omega(1)}$. If N is the size of the graph produced in the FGLSS reduction, then $N = 2^{r+f} = 2^{O(\log n)} = n^{O(1)}$. Thus we get a hardness of N^c for some constant $c > 0$.

3 Amortized free bits and hardness of $n^{1-\epsilon}$

DEFINITION 5 *For a language $L \in PCP_{c,s}[r,q,f]$, we define its amortized free bit complexity to be:* $\bar{f} = \frac{f}{\log 1/s}$.

The above definition assumes that the completeness parameter $c \approx 1$. In general, one could define $\bar{f} = \frac{f+\log 1/c}{\log 1/s}$. In this section we will obtain hardness results as a function of $\bar{f}$. For that we will need to construct graphs of the following form:

LEMMA 8
For any $R, r < R$, there exists a bipartite graph $G = (V, W, E)$ with $|V| = 2^R$, $|W| = 2^r$, $deg(v) = k = \frac{R+2}{\log 1/s}$ $\forall v \in V$ such that for any $W' \subseteq W$, $|W'| = s|W|$,

$$\mathbf{Pr}_v[N(v) \subseteq W'] \leq \frac{2^r}{2^R} \tag{2}$$

Such graphs are called *dispersers* and the proof of their existence is based on the probabilistic method [2]. The idea for the probabilistic construction of dispersers is due to Zuckerman [8]. We can now use this construction to obtain the following hardness result, proved in [1].

THEOREM 9
If $SAT \in PCP_{c,s}[r,q,f]$, then MAX-CLIQUE is $n^{\frac{1}{1+\bar{f}}-\epsilon}$-hard, unless $\mathbf{NP} = \mathbf{BPP}$.

PROOF: Let X be a verifier for SAT with parameters c, s, r, q, f. We can construct a new verifier Y just as we did in Theorem 7 but by using dispersers instead of the graphs that we used in Theorem 7. By using Lemma 8, we can see that the parameters c', s', r', q', f' of the new verifier will be:

1. $c' \geq c^k/2$ (here we need the property that for *some fixed* subset $W' \subseteq W, |W'| = c|W|$, we have $\Pr_v[N(v) \subseteq W'] \geq c^k/2$. This holds because the disperser is constructed by choosing for every $v \in V$, k random neighbors in W).
2. $s' \leq 2^r/2^R$
3. $r' = R$
4. $f' = kf$
5. $q' = kq$

The hardness factor that we establish by using Theorem 1 is then $\frac{c'}{s'} = \frac{c^k 2^R}{2^{r+1}}$. The size of the graph that we obtain is $N = 2^{r'+f'}$. We want to express the hardness factor as a power of N, i.e., we want to find an α such that $N^\alpha = \frac{c^k 2^R}{2^{r+1}}$. If we take the logarithm on each side and substitute $\log N = r' + f'$, we have

$$\alpha = \frac{R - r - 1 - k\log 1/c}{R + kf} \geq \frac{R}{R+kf} - \frac{k\log 1/c + r + 1}{R}$$

If $\frac{r}{R} = \epsilon$ and we substitute the value of k from Lemma 8, we finally obtain a hardness result of $N^{\frac{1}{1+\bar{f}}-\epsilon}$. $\square$

REMARK 1 The reduction of Theorem 9 is a randomized reduction, since the construction of a disperser in Lemma 8 is probabilistic.

In oder to prove hardness of $n^{1-\epsilon}$ for MAX-CLIQUE, it suffices to show that every language in **NP** has a PCP verifier with $\bar{f} \to 0$. We cannot show this by running repeatedly an existing PCP verifier. If we run k times a verifier, with parameters f and s, the new free bit complexity will be $\frac{kf}{\log 1/s^k} = \frac{f}{\log 1/s}$, which is the same as in the original verifier. Hence, repetitions do not help to reduce the amortized free bit complexity.

The first PCP construction with $\bar{f} \to 0$ was obtained by Hastad [4]. Here we will present a different construction by Samorodnitsky and Trevisan [7], which greatly simplified the result of [4]. An even simpler proof was also obtained in [5].

4 Samordnitsky-Trevisan's PCP

We will only state the PCP construction of [7], which we will call ST PCP. It works by *iterating* Håstad's 3-bit PCP via a clever *recycling of queries.*

Let us briefly recall Håstad's 3-bit PCP (see Lectures 4 and 5). There we transform an instance of an **NP** language to a label cover instance $\mathcal{L}(G(V,W,E),[M],[N],\{\pi_{vw}\}_{(v,w)\in W})$. The proof is supposed to be the encodings of labels of all $v \in V$ and $w \in W$. The verification process works as follows : we pick an edge $(v,w) \in E$ at random from the bipartite graph of the label cover instance. Let A and B be the supposed encodings for labels of v and w respectively. We pick a random position x from the table A, and a random position y from the table B. We then test if $B(z) = A(x)B(y)$ where $z = y(x \circ \pi)\mu$ for some perturbation μ.

The verifier of the ST PCP will do the following:

- Pick a vertex v at random from the left hand side of the graph corresponding to the label cover instance. Pick locations $x_1, ..., x_t$ at random from table A (the supposed encoding of label of v).
- Pick t neighbors $w_1, ..., w_t$ of v and a location y_j at random from table B_j (the supposed encoding of label of w_j).
- Run the 3-bit PCP on (x_i, y_j), $1 \leq i,j \leq t$ and accept if all the t^2 runs are accepted. To be precise, define $z_{ij} = y_j(x_i \circ \pi_j)\mu_{ij}$ and accept iff

$$B_j(z_{ij}) = A(x_i)B_j(y_j) \qquad \forall 1 \leq i,j \leq k$$

The verifier makes t^2 linearity tests. In [7] it is shown that the ST PCP achieves the following parameters:

1. $c \approx 1$, because Hastad's PCP has almost perfect completeness.
2. $s \leq 2^{-t^2} + \eta$, where η is arbitrarily small.
3. $f = 2t$, because $A(x_i)$ and $B_j(y_j)$ are free. The queries $B_j(z_{ij})$ are non-free.

Therefore, the amortized free bit complexity of the ST PCP is $\frac{f}{\log 1/s} = \frac{2t}{t^2} \to 0$. The surprising property of the ST PCP is its soundness parameter. Even though the t^2 linearity tests heavily share query-bits, these tests behave as if they were independent. Each test cuts down the soundness by a factor $1/2$, achieving soundness essentially 2^{-t^2}.

COROLLARY 10
For any $\epsilon > 0$, there is no $n^{1-\epsilon}$ approximation algorithm for MAX-CLIQUE, unless **NP** = **BPP**.

The FGLSS reduction can be used to prove hardness results for other optimization problems. In particular, by combining Hastad's 3-bit PCP and the FGLSS reduction we can show a hardness result for Vertex Cover:

THEOREM 11
There is no $(7/6-\epsilon)$-factor approximation algorithm for Vertex Cover, unless **P** = **NP**.

PROOF: Given an **NP**-hard language L, we know by the 3-bit PCP construction that there is a PCP verifier for L with $f = 2, c \approx 1-\epsilon, s = 1/2+\epsilon$. Given an instance x, we apply the FGLSS reduction to construct the graph G_x. If $x \in L$, then there is an independent set I with $|I| \geq c2^{-f}|G_x|$. But then the remaining vertices form a vertex cover, hence there exists a vertex cover of size at most $|G_x|(1-c2^{-f})$. On the other hand if $x \notin L$, it is also easy to see that any vertex cover has size at least $|G_x|(1-s2^{-f})$. Therefore we obtain a hardness of $\frac{1-s2^{-f}}{1-c2^{-f}} \approx 7/6$. □

Improving the hardness of vertex cover to $2-\epsilon$ is equivalent to the following characterization of **NP**.

THEOREM 12
NP *has a PCP verifier with $f = 0, c = 1/2-\epsilon, s \to 0$ iff Vertex Cover is hard to approximate within $2-\epsilon$.*

References

[1] M. Bellare, O. Goldreich, M. Sudan. Free bits, PCP's and non-approximability - towards tight results. *SIAM Journal on Computing*, 27(3), 804–915, 1998. Preliminary version in *FOCS '95*.

[2] L. Engebretsen, J. Holmerin. Towards Optimal Lower Bounds for Clique and Chromatic Number. *Theoretical Computer Science*, 2003.

[3] U. Feige, S. Goldwasser, L. Lovász, S. Safra, M. Szegedy. Interacive Proofs and the Hardness of Approximating Cliques. *Journal of the ACM*, 43(2), 268–292, 1996.

[4] J. Hastad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[5] J. Hastad, A. Wigderson. Simple analysis of graph tests for linearity and PCP. *IEEE Conference on Computational Complexity*, 244–254, 2001.

[6] M. Luby, A. Wigderson. Pairwise Independence and Derandomization. ICSI TR-95-035, 1995.

[7] A. Samorodnitsky, L. Trevisan. A PCP Characterization of NP with Optimal Amortized Query Complexity. *STOC*, 2000.

[8] D. Zuckerman. On Unapproximable Versions of **NP**-complete Problems. *SIAM Journal on Computing*, 25(6), 1293–1304, 1996.