CMPUT 675: Approximation Algorithms **Fall 2014**

Lecture 31 (Nov 26): Maximum Independent Set Hardness

Lecturer: Zachary Friggstad *Scribe: Zachary Friggstad*

31.1 Constant-Factor Hardness

Our ultimate goal will be to show that there is some constant δ such that there is no $1/n^\delta$ approximation for Maximum Independent Set unless $\mathbf{P} = \mathbf{NP}$ where n is the number of nodes in the graph. We start by proving a weaker result, after one important definition.

Definition 1 *Consider an $(r(n), q(n))$-restricted verifier V. For $b \in \{0,1\}^{r(n)}, y \in \{0,1\}^{q(n)}$ and a proof string π, say (b, y) agrees with π if $\pi_{i^b_j} = y_j$ for each $1 \leq j \leq q(n)$ where $i^b_1, \ldots, i^b_{q(n)}$ are the indices of π that will be queried when V is supplied with random bit string b.*

For $b, b' \in \{0,1\}^{r(n)}, y, y' \in \{0,1\}^{q(n)}$, say that (b, y) agrees with (b', y') if there is some proof string π such that both (b, y) and (b', y') agree with π.

In other words, we think of the tuple (b, y) as being an actual assignment to the proof string bits that will be queried when V is given random string b. Saying (b, y) agrees with a proof string π simply means that the contents of π at the queried bits are exactly the bits in the y-vector.

Important Observation
Note that if S is a subset of pairs (b, y) such that every two agree according to Definition 1, then there is some proof string π such that every $(b, i) \in S$ agrees with π. In particular, the string π defined by

$$\pi_i = \begin{cases} y_{i^b_j} & \text{if some } (b,y) \in S \text{ is such that } i^b_j = i \text{ for some } j \\ 0 & \text{otherwise} \end{cases}$$

The first part of the definition of π is independent of the choice $(b, y) \in S$ (if there is more than one) because no two $(b, y), (b', y') \in S$ disagree on any bit of the proof. The value 0 in the second part of the definition is arbitrary.

Theorem 1 *For any $c > 1/2$, there is no c-approximation for* Maximum Independent Set *unless* $\mathbf{P} = \mathbf{NP}$.

Recall that in assignment 1 you were asked to show that if there is an α-approximation for some constant α, then there is in fact a $\sqrt{\alpha}$-approximation. A corollary of Theorem 1 is that there is no constant-factor approximation for the Maximum Independent Set problem unless $\mathbf{P} = \mathbf{NP}$, otherwise we could apply the above "$\alpha \Rightarrow \sqrt{\alpha}$" result a constant number of times to this constant-factor approximation to get a c'-approximation for some $c' > 1/2$.

Proof. Let $L \in \mathbf{NP}$ and, by the PCP Theorem, let V be a $(r \cdot \log_2 n, q)$-restricted verifier with completeness 1 and soundness $1/2$ for L where r, q are constants.

In polynomial time, we will reduce an instance of the decision problem $x \in L$? to a graph $G = (U, E)$ with $|U| = |x|^r \cdot 2^q$ such that:

- **Completeness**: If $x \in L$ then G has an independent set of size $|x|$.
- **Soundness**: If $x \notin L$, then the largest independent set of G has size at most $|x|/2$.

This $|x|$ vs. $|x|/2$ hardness gap shows we cannot approximate MAXIMUM INDEPENDENT SET within any factor better than $1/2$.

The reduction is quite simple. We let

$$U = \{v_{b,y} : b \in \{0,1\}^{r \cdot \log_2 |x|}, y \in \{0,1\}^q\}.$$

Here, it is useful to associate $v_{b,y} \in U$ with the random bit string b and $y = (y_1, y_2, \ldots, y_q)$ with a particular setting of the bits at positions $i_1^b, i_2^b, \ldots, i_q^b$ of the proof string that will be queried by verifier V when supplied with random bit string b.

Let $U_{good} \subseteq U$ be the nodes $v_{b,y}$ such that $V(x, b, \pi) =$ ACCEPT for some proof string π that agrees with (b, y). This is the same as saying $V(x, b, \pi) =$ ACCEPT for *any* proof string π that agrees with (b, y) because the computation of $V(x, b, \pi)$ is not affected by the bits of π it does not query.

The edges of G are defined as follows:

$$E = \{(v_{b,y}, v_{b',y'}) : |\{v_{b,y}, v_{b',y'}\} \cap U_{good}| \leq 1 \text{ or } (b, y) \text{ does not agree with } (b', y')\}.$$

Note that $|U| = |x|^r \cdot 2^q$ (as promised) which is polynomial in $|x|$. Furthermore, we can construct E in polynomial time by simulating V (in polynomial time) on each (b, y) pair to see if $v_{b,y} \in U_{good}$. The predicate (b, y) agrees with (b', y') can also be checked efficiently, simply see if they try to assign different values to the same position of the proof string.

Completeness
We claim that G has an independent set of size $|x|^r$. To see this, let $\overline{\pi}$ be a proof string such that $V(x, b, \pi) =$ ACCEPT for any random bit string b. Consider

$$W = \{v_{b,y} : (b, y) \text{ agrees with } \overline{\pi}\}$$

Clearly for every b there is precisely one y such that (b, y) agrees with $\overline{\pi}$, so $|W| = 2^{r \cdot \log_2 |x|} = |x|^r$.

Furthermore, we claim that W is an independent set. Consider any two $v_{b,y}, v_{b',y'} \in W$. Both are in U_{good} and agree with each other because $V(x, b, \overline{\pi}) = V(x, b', \overline{\pi}) =$ ACCEPT and $\overline{\pi}$ agrees with both (b, y) and (b', y'). So, $(v_{b,y}, v_{b',y'}) \notin E$.

Soundness
Consider any independent set W' of G. If $W' - U_{good} \neq \emptyset$ then $|W'| = 1$ because there is an edge between $v_{b,y}$ and every other vertex in U if $v_{b,y} \notin U_{good}$. So, we assume that $W' \subseteq U_{good}$.

For each $b \in \{0,1\}^{r \cdot \log_2 n}$ there is at most one $y \in \{0,1\}^q$ such that $v_{b,y} \in W'$ because (b, y) does not agree with (b, y') for any $y \neq y'$. Let

$$B_{W'} = \{b : v_{b,y} \in W' \text{ for some } y\}$$

and note $|B_{W'}| = |W'|$.

Now, any two $(b, y), (b', y')$ such that $v_{b,y}, v_{b',y'} \in W'$ must agree (because W' is an independent set) so, by the important observation preceding the theorem statement, there is some proof string π' that agrees with every (b, y) such that $v_{b,y} \in W'$.

Let $B' = \{b : V(x, b, \pi') = \text{ACCEPT}\}$. We have $\Pr[V(x, b, \pi') = \text{ACCEPT}] \leq 1/2$ (when randomly choosing b), which means $|B'| \leq |x|^r/2$ (i.e. B' consists of at most half of the possible random bit strings).

Finally, we have $B_{W'} \subseteq B'$ because $W' \subseteq U_{good}$ (so every string in $B_{W'}$ causes V to accept π'). This means

$$|W'| = |B_{W'}| \leq |B'| \leq |x|^r/2.$$

■

Polynomial Hardness

We now focus on proving the stronger statement that there is no $1/n^\delta$-approximation for some constant $\delta > 0$. Our starting point is yet another alternative characterization of **NP**.

Theorem 2 *There are constants r', q' such that $\mathbf{NP} = \mathbf{PCP}_{1,1/n}(r' \cdot \log_2 n, q' \cdot \log_2 n)$.*

We will prove this in the next section, but let's see how it can be used to get polynomial hardness for maximum independent set.

Theorem 3 *There is a constant $\delta > 0$ such there is no approximation algorithm with approximation guarantee better than $1/n^\delta$ unless $\mathbf{P} = \mathbf{NP}$.*

Proof. Start with the $(r' \cdot \log_2 n, q' \cdot \log_2 n)$-restricted verifier V' for a language L in **NP** with completeness 1 and soundness $1/n$ (as promised by Theorem 2).

From here, use the exact same reduction as in the proof of Theorem 1 to get a graph $G = (U, E)$ where $U = \{v_{b,y} : b \in \{0,1\}^{r' \cdot \log_2 n}, y \in \{0,1\}^{q' \cdot \log_2 n}\}$ and

$$E = \{(v_{b,y}, v_{b',y'}) : |\{v_{b,y}, v_{b',y'}\} \cap U_{good}| \leq 1 \text{ or } (b, y) \text{ does not agree with } (b', y')\}.$$

In this case, we have $|U| = 2^{r' \cdot \log_2 |x|} \cdot 2^{q' \cdot \log_2 |x|} = |x|^{r'+q'}$, which is polynomial in $|x|$. Overall, the construction of this graph G still takes poly($|x|$) time.

We sketch the essential differences in the analysis between this reduction and the reduction in Theorem 1.

Completeness
Defining W in the same way again yields an independent set of size $|x|^r$.

Soundness
For any independent set W', we define $B_{W'}$ and B' as in the proof of Theorem 1. The main difference is that $|B'|$ is only a $1/|x|$-fraction of all possible random bit strings because the soundness in the verifier V' is the much smaller value $1/|x|$. This allows us to conclude the stronger statement

$$|W'| = |B_{W'}| \leq |B'| \leq |x|^r/|x| = |x|^{r-1}.$$

Putting It Together
This introduces an $|x|^r$ vs. $|x|^{r-1}$ gap, so there is no $c$approximation for any $c > |x|^{r-1}/|x|^r = 1/|x|$. However,

we want to state this hardness in terms of the new graph G, which has $n := |U| = |x|^{r'+q'}$ vertices. Let $\delta = 1/(r'+q')$ and note that δ is a constant. Then $|x| = n^\delta$, so we cannot approximate MAXIMUM INDEPENDENT SET better than $1/n^\delta$ for this constant δ. ■

31.2 Using Expanders to Amplify Hardness

In this section, we prove Theorem 2. Actually, we only prove the following direction.

Theorem 4 *For some constants q', r', we have* $\mathbf{NP} \subseteq \mathbf{PCP}_{1,1/n}(r' \cdot \log_2 n, q' \cdot \log_2 n)$.

The other direction is simple and is just briefly sketched here. Since the number of queried bits is $O(\log n)$, then we can still use such a verifier to produce, in polynomial time, an instance of SAT that is satisfiable in the completeness case and is not satisfiable in the soundness case using essentially the same reduction from the PCP Theorem to an instance of MAX-qSAT from the last lecture. This gives a poly-time reduction from any language $L \in \mathbf{PCP}_{1,1/n}(r' \cdot \log_2 n, q' \cdot \log_2 n)$ to SAT, so $L \in \mathbf{NP}$.

Before proving Theorem 4, we briefly discuss its intuition. If we have a verifier as in the PCP Theorem, we can reduce the soundness from $1/2$ to $1/n$ simply by executing $\log_2 n$ sequential repetitions of V on the supplied proof π, rejecting if even one repetition caused V to reject. Here, we use a new sequence of $r \cdot \log_2 n$ random bits.

In the completeness case, V accepts some π with probability 1 so this same π will cause V to accept in each of these $\log_2 n$ repetitions. In the soundness case, V accepts any π only with probability at most $1/2$ so $k := \log_2 n$ sequential iterations will have all runs accepting π with probability only $1/2^k = 1/n$. This is almost what we need, as the number of queried bits will be $q \cdot k = q \cdot \log_2 n$.

Unfortunately, running the reduction in Theorem 3 from a verifier that reads $\Theta(\log^2 n)$ random bits will produce a graph $G = (U, E)$ with $|U| = |x|^{\Theta(\log^2 |x|)} = |x|^{\Theta(\log |x|)}$ which is not a polynomial-time reduction! To reduce the number of random bits to only logarithmic while maintaining $1/n$ soundness requires one additional tool.

Theorem 5 (and Definition) *There is some constant d such that for every n, there is graph $G_n = (U_n, E_n)$ (with, perhaps, parallel edges) such that*

- *Every vertex of G_n has degree exactly d.*
- *For every $S \subseteq U_n$ with $|S| \leq n/2$, $|\delta(S)| \geq |S|$.*

Furthermore, G_n can be constructed in poly(n)*-time.*

Such graphs are called *expander graphs* (see [HLW06] for a more general definition).

These are rather remarkable graphs. One one hand they are very sparse since each vertex has constant degree. On the other hand, they are very well connected in the sense that in each cut of the graph where S is the smaller side, the number of edges crossing S is a constant fraction of the number of edges contained in S. In some sense, this means a random walk cannot stay contained in a small set S for very long.

Let's formalize this. Say that a *length k* random walk in a graph G is a sequence of nodes $v_1, v_1, \ldots, v_k$ generated according to the following random process.

- The first vertex v_1 is chosen uniformly at random from the nodes of G.

- Iteratively for each $2 \leq i \leq k$, we choose v_i uniformly at random from the neighbours of v_{i-1}.

A random walk in an expander has nice properties that seem similar to simply independently sampling each v_i from the set of all nodes. That is, if $|S| \leq n/2$ then randomly choosing the v_i uniformly among all nodes (not just neighbours of the previous nodes) will have all $v_i \in S$ with probability at most $1/2^k$. In an expander, since a constant-fraction of the edges with an endpoint in S leave S then one would hope that we would see similar bounds on the probability that a random walk stays in S. We do.

Theorem 6 (Expander Walks) *Consider a random walk $v_1, \ldots, v_k$ in one of the G_n expander graphs defined above. For any subset $S \subseteq U_n$ with $|S| \leq n/2$, we have*

$$\Pr[v_i \in S \text{ for each } 1 \leq i \leq k] \leq 2^{-k/c}$$

where c is some universal constant.

Expander walks are precisely what we need. We will "overlay" an expander on the set of random bit strings and perform an random walk of length $c \cdot \log_2 n$ in this graph, running V with every random bit string/vertex we see. This will query $O(\log n)$ positions of the proof string $O(1)$ for each of the $O(\log n)$ random bit strings seen, it will stay within the "bad" set of accepting random bit strings with probability at most $1/n$ in the soundness case, but the number of random bits required to do this walk is only $O(\log n)$ as opposed to $O(\log^2 n)$ because each step after sampling the first vertex v_1 requires only $\log_2 d$ bits.

Proof of Theorem 4. Let L be a language in **NP**. By the PCP Theorem, there is an $(r \cdot \log_2 n, q)$-restricted verifier V for L with completeness 1 and soundness $1/2$ where r, q are constants. We use V to get an $(r' \cdot \log_2 n, q' \cdot \log_2 n)$-restricted verifier V' for L with completeness 1 and soundness $1/n$ where r', q' are constants.

Let $N = |x|^r$ be the number of possible random bit strings of length $r \cdot \log_2 |x|$. Construct the expander graph $G_N = (U_N, E_N)$ from Theorem 5, which can be done in poly($|x|$) time. We identify U_N with the set $\{0,1\}^{r \cdot \log_2 |x|}$ in any arbitrary way, so it is useful to think of U_N as the set of possible random bit strings of length $r \cdot \log_2 |x|$.

Let c be the constant from Theorem 6 and let $k = c \cdot \log_2 |x|$. Our final verifier V' simply runs the verifier V with k different random bit strings. But the random bit strings are not chosen independently in each run. Rather, we let $b_1, \ldots, b_k \in U_N$ be random bit strings that we construct from a length k random walk in G_N: choose $b_1 \in U_N$ uniformly at random and then for each $i \geq 2$ we choose b_i randomly among the d neighbours of b_{i-1}.

Verifier V' accepts the proof π of the statement $x \in L$ if and only if $V(x, b_i, \pi) =$ accept for each $1 \leq i \leq k$.

Completeness
If $x \in L$, then there is some π such that $V(x, b, \pi) =$ ACCEPT for every random bit string b. Then with this same proof string π, we will have $V(x, b_i, \pi)$ for any sequence of random bit strings $b_1, \ldots, b_k$ generated by the random walk on G_N, so V' accepts π with probability 1.

Soundness
Consider any proof string π and let $U_{bad} = \{b \in U_N : V(x, b, \pi) =$ ACCEPT$\}$. By the soundness of V, $|U_{bad}| \leq |U_N|/2$. By Theorem 6, the probability that $b_i \in U_{bad}$ for each b_i generated in length k random walk is at most $2^{-k/c} = 1/|x|$. That is, V' accepts π with probability at most $1/|x|$.

Size of the Parameters
The total number of bits of π that are queries is exactly $k \cdot q = c \cdot q \cdot \log_2 |x|$, so we let $q' = c \cdot q$ which is a constant. The random walk can be generated using only $O(\log |x|)$ random bits. In particular, $r \cdot \log_2 |x|$ random bits are used to sample the first vertex/bit string b_1. Then since b_i is a random neighbour of b_{i-1} for $i \geq 2$, we can sample b_i given b_{i-1} using only $\log_2 d$ bits. Overall, the number of bits used is at most $r \cdot \log_2 n + (k-1) \cdot \log_2 d \leq (r + c \cdot \log_2 d) \cdot \log_2 n$ so we let $r' = r + c \cdot \log_2 n$ which is also constant.

We have just shown V' is an $(r' \cdot \log_2 n, q' \cdot \log_2 n)$-restricted verifier for L with completeness 1 and soundness $1/n$. Also, it is easy to see that V' runs in time poly($|x|$). Therefore, $L \in \mathbf{PCP}_{1,1/n}(r' \cdot \log_2 n, q' \cdot \log_2 n)$. ■

31.3 Discussion

The first inapproximability results for the Maximum Independent Set problem predate the PCP theorem [F+91]. Håstad shows a much stronger result, for any constant $\epsilon > 0$ there is no $1/n^{1-\epsilon}$-approximation unless $\mathbf{NP} = \mathbf{ZPP}$ [H99] ($\mathbf{ZPP}$ is the class of languages that can be decided with a randomized algorithm has polynomial *expected* running time and always returns the correct solution). This was refined by Zuckerman, who proved the same hardness bound under the more standard assumption $\mathbf{P} \neq \mathbf{NP}$ [Z07]. Under even stronger (but still plausible) assumptions, Khot and Ponnuswami show that in fact it is hard to approximate better than $(2^{\log^{3/4+\epsilon'}(n)})/n$ for any constant $\epsilon' > 0$, which is asymptotically smaller than $1/n^{1-\epsilon}$ for any constant $\epsilon > 0$.

On the positive side, Feige showed that we can approximate the Maximum Independent Set problem within a factor of $\Omega(\log^3 n/(n \cdot (\log \log n)^2))$ [F04], which is slightly better than the trivial $1/n$-approximation that outputs a single vertex.

As suggested by these notes, expander graphs are very useful in proving hardness-of-approximation results. In fact, they play a pivotal role in Dinur's simplified proof of the PCP Theorem itself [D07]. An excellent introduction can be found in an article by Hoory, Linial, and Wigderson [HLW06], and also in the text by Arora and Barak.

References

D07 I. Dinur, The PCP theorem by gap amplification, Journal of the ACM, 54(3):12, 2007.

F04 U. Feige, Approximating maximum clique by removing subgraphs, SIAM Journal on Discrete Mathematics, 18(2):219–225, 2004.

F+91 U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy, Approximating clique is almost NP-complete, In Proceedings of FOCS, 1991.

H99 J. Håstad, Clique is hard to approximate within $n^{1-\epsilon}$, Acta Mathematica, 182:105–142, 1999.

HLW06 S. Hoory, N. Linial, and A. Wigderson, Expander graphs and their applications, Bulletin of the American Mathematical Society, 43:439–561, 2006.

Z07 D. Zuckerman, Linear degree extractors and the inapproximability of max clique and chromatic number, Theory of Computing, 3:103–128, 2007.