CMSC 39600: PCPs, codes and inapproximability 27 Sep, 2007

Lecture 2: PCPs – definitions and inapproximability of clique

Lecturer: Prahladh Harsha *Scribe: Karthik Sridharan*

In today's lecture, we will formally define the PCP classes, state the PCP Theorem, its various strenghtenings and finally prove the NP-hardness of approximating the clique.

2.1 PCP Classes – Definition

We first define a restricted (probabilistic) verifier which on input a statement x, probes a proof at a few randomly selected positions and then accepts or rejects the proof. Based on this verifier we shall then define the PCP class.

Definition 2.1 (restricted verifier). *Let $r, q, m, t : \mathbb{N} \to \mathbb{N}$ be integer valued functions and Σ an alphabet. A $(r, q, m, t)_\Sigma$-restricted verifier V is a probabilistic Turing Machine (TM) with oracle access to a proof π over the alphabet Σ, which on input x of length n.*

- *tosses at most $r(n)$ coins*
- *probes at most $q(n)$ locations of*
- *a proof π of size at most $m(n)$*
- *runs in time $t(n)$*
- *and based on the proof bits it reads, it either accepts or rejects the proof.*

We will denote the verdict of the verifier V on input x and proof π and random coins R by $V^\pi[x; R]$.

Definition 2.2 (PCP Classes). *Let $r, q, m, t : \mathbb{N} \to \mathbb{N}$ be integer valued functions and Σ an alphabet. We say that a language $L \in PCP^\Sigma_{c,s}[r, q, m, t]$ if L has a $(r, q, m, t)_\Sigma$-restricted verifier V such that*

Completeness: $\forall x \in L, \exists \pi$ *of size at most* $m(|x|), \Pr_R\left[V^\pi[x; R] = \mathsf{acc}\right] \geq c(n)$.

Soundness: $\forall x \notin L, \forall \pi$ *of size at most* $m(|x|) \Pr_R[V^\pi[x; R] = \mathsf{acc}] \leq s(n)$.

If the alphabet is the binary alphabet $\{0, 1\}$, we will usually omit it. Similarly, we will omit mentioning the running time $t(n)$ if $t(n) = \Omega(\text{poly}(n))$ and the proof length $m(n)$ of the proof if $m(n) = \Omega(2^{r(n)+q(n)})$.

Remark

- If $c(n) = 1$, we say that the PCP verifier has perfect completeness.
- The verifier could be either non-adaptive or adaptive (i.e., the locations probed by the verifier could depend on the earlier probes). To explicitly specify that a particular PCP class is obtained by an adaptive (non-adaptive verifier), we will use the notation a-PCP (na-PCP).
- If the verifier is non-adaptive then the size of the proof is bounded above by $q(n)2^{r(n)}$ while if the verifier is adaptive, then $m(n) \leq 2^{r(n)+q(n)}$.
- The number of queries is bounded by the running time (i.e., $q(n) \leq t(n)$), which could be as large as polynomial in the length of the input.

- $PCP_{c,s}[r,q] \subseteq NTIME(2^{r(n)+q(n)})$: The non-deterministic verifier guesses the proof of length 2^{r+q} and runs the PCP verifier for all possible random coins and accepts if the accepting probability is at least $c(n)$.
- It follows from definition that $NP = PCP_{1,0}[0, \text{poly}(n)], BPP = PCP_{\frac{2}{3},\frac{1}{3}}[\text{poly}(n), 0], P = PCP_{1,0}[0,0]$.

2.2 PCP Theorems

The PCP Theorem can now be stated using th above notation as follows:

Theorem 2.3 (PCP Theorem [AS98, ALM+98])**.**

$$\exists q > 0 : NP = \bigcup_{c>0} PCP_{1,\frac{1}{2}}[c \log n,\ q]$$

Note here that the inclusion $PCP_{1,\frac{1}{2}}[\log n,\ q] \subseteq NP$ follows from the above remark that $PCP[r,q] \subseteq NTIME(2^{r+q})$. The PCP Theorem proves the inclusion in the opposite direction.

2.2.1 A (Brief) History of PCP Theorem

The study of probabilistic checking of proof was initiated in the independent works of Babai and Moran [BM88]; and Goldwasser, Micali and Rackoff [GMR89]. Following a sequence of results, Babai, Fortnow and Lund [BFL91] and Fortnow, Rompel and Sipser [FRS94] proved the following exponential version of the PCP Theorem:

$$NEXP = PCP_{1,\frac{1}{2}}[\text{poly}(n), \text{poly}(n)].$$

Feige, Goldwasser, Lovász, Safra and Szegedy [FGL+96] and independently Babai, Fortnow, Levin and Szegedy [BFLS91] scaled down the above result for non-deterministic exponential time to non-deterministic polynomial time with different motivations. Babai et al. [BFLS91], showed that any proof can be efficiently (re)written in such a manner that the rewritten proof can be checked with at most $\text{poly}\log n$ probes.

$$NP \subseteq PCP_{1,\frac{1}{2}}[r(n) = \text{poly}\log(n), q(n) = \text{poly}\log(n), m(n) = \text{poly}(n), t(n) = \text{poly}\log(n)].$$

Feige et al. [FGL+96] proved the following:

$$NP \subseteq PCP_{1,\frac{1}{2}}[\log n \log\log n, \log n \log\log n]$$

and established the dramatic connection between PCPs and hardness of approximation. We will discuss this reduction of [FGL+96] showing the hardness of approximating clique later in today's lecture. Subsequently, Arora and Safra [AS98] showed that $NP = PCP_{1,\frac{1}{2}}[\log(n), \sqrt{\log(n)}]$ and immediately after, Arora, Lund, Motwani, Sudan and Szegedy [ALM+98], showed that the number of queries can be brought down to a constant, giving the PCP Theorem.

In this course, we will not follow the above sequence of results to prove the PCP Theorem. We will instead give the recent proof of the PCP Theorem due to Dinur [Din07].

2.2.2 Strengthenings of the PCP Theorem

We shall see various strengthening of the PCP Theorem in this course. The first strengthening, called the parallel repetition theorem, by Raz, shows that the error probability can be brought down arbitrarily if we allow the alphabet to grow polynomially in the inverse of the error.

Theorem 2.4 (Parallel Repetition Theorem [Raz98])**.** *For any $\epsilon > 0$, there exists alphabet Σ such that $|\Sigma| = \mathrm{poly}(\frac{1}{\epsilon})$ and*

$$NP \subseteq \bigcup_{c>0} PCP^{\Sigma}_{1,\epsilon}[c \log n, 2]$$

Using the parallel repetition theorem, Håstad then showed that the constant q in the PCP Theorem can be brought down to 3.

Theorem 2.5 ([Hås01])**.** *For any $\epsilon > 0$,*

$$NP \subseteq \bigcup_{c>0} PCP_{1-\epsilon,\frac{1}{2}+\epsilon}[c \log n, 3]$$

Furthermore, the actions of the above PCP verifier operation is very simple: It merely queries the proof in three locations (i_1, i_2, i_3) based on its random coins and accepts iff $\pi_{i_1} \oplus \pi_{i_2} \oplus \pi_{i_3} = b$.

The following optimal inapproximability result for MAX-3SAT follows from the above Håstad's Theorem. Note that for a random assignment satisfies 7/8 fraction of the clauses. Thus, a trivial randomized algorithm achieves 7/8-approximation for MAX-3SAT (this algorithm can be derandomized). The following corollary states that to do any better, would imply NP=P!

Corollary 2.6. $\forall \epsilon > 0$*, it is NP-Hard to approximate MAX-3SAT to within* $\frac{7}{8} + \epsilon$

Note that the above 3-query PCP does not have perfect completeness. In fact, it is known that any non-adaptive 3-query PCP for NP with perfect completeness cannot achieve soundness better than 5/8 unless NP=P. Later, Guruswami, Lewin, Sudan, and Trevisan [GLST98] constructed an adaptive 3-query PCP for NP with perfect completeness and soundness arbitrarily close to 1/2.

Theorem 2.7 ([GLST98])**.** *For any $\epsilon > 0$,*

$$NP \subseteq \bigcup_{c>0} a\text{-}PCP_{1,\frac{1}{2}+\epsilon}[c \log n, 3]$$

The above strengthenings of the PCP Theorem though obtain optimal results with respect to query complexity and soundness behave very badly with respect to proof size. In fact, it is reputed that the proof size $m(n)$ in Håstad's Theorem is of the order of n^{10^6}, a polynomial, albeit a large one. There has been progress in the orthogonal direction of shortening the proof size starting from the work of Babai et al [BFLS91], resulting in the following PCP due to Ben-Sasson and Sudan [BS05] and Dinur [Din07]

Theorem 2.8 ([BS05, Din07])**.**

$$NP \subseteq \bigcup_{c>0} PCP_{1,\frac{1}{2}}[r(n) = \log n, q(n) = 3, m(n) = n\mathrm{poly}\log n]$$

2.3 Hardness of Approximating Clique

We will now assume the PCP Theorem and prove the NP-hardness of approximating the MAX-CLIQUE problem. For this, we first recall the approximate decision problem gap-CLIQUE$_\alpha$, defined in the first lecture.

Definition 2.9 (gap-CLIQUE)**.** *The instance of gap-CLIQUE$_\alpha$ (for each $0 < \alpha \leq 1$)are of the form $\langle G, k\rangle$, where G is a graph and k a positive integer. The YES and NO instances of gap-CLIQUE$_\alpha$ are define*

$$\begin{aligned} YES &= \{\langle G, k\rangle | CLIQUE(G) \geq k\} \\ NO &= \{\langle G, k\rangle | CLIQUE(G) \leq \alpha k\} \end{aligned}$$

where $CLIQUE(G)$ denotes the size of the largest clique in G.

We will prove the following reduction (due to Feige et al. [FGL$^+$96]) which will prove the NP-hardness of gap-CLIQUE$_\alpha$ for some $0 < \alpha < 1$, which in turns proves the NP-hardness of approximating MAX-CLIQUE to a factor better than α.

Lemma 2.10 ([FGL$^+$96])**.** *If 3-COLOR $\in PCP_{c,s}[r, q]$ then there exists a deterministic reduction running in time* $\mathrm{poly}(2^{r+q})$ *reducing 3-COLOR to gap-CLIQUE$_{s/c}$.*

Proof. Consider a PCP verifier Ver for 3-COLOR that shows 3-COLOR $\in PCP_{c,s}[r, q]$. We use this verifier to reduce an instance H of 3-COLOR to an instance $\langle G, k\rangle$ of gap-CLIQUE$_{s/c}$. The basic idea is to encode the actions of the PCP verifier Ver by the graph G such that if $H \in$ 3-COLOR then G has a clique of size at least k and if $H \notin$ 3-COLOR then G does not have any clique of size greater than $(s/c)k$ for some k.

What are the actions of the PCP Verifier Ver? On input the graph H, it tosses random coins R (uniformly at random of 2^r possibilities). Based on these random coins R, the verifier decides to probe the proof at q locations $(i_1(R), \ldots, i_q(R))$. It then probes the proof π at these locations to obtain $(\pi_{i_1(R)}, \pi_{i_2(R)}, \ldots, \pi_{i_q(R)})$. We call this sequence of q bits, the "view" of the verifier. Note that there are exactly 2^q possible views for each random coin (depending on the proof). Some of these views are accepting (i.e., they cause the verifier to accept) while others are rejecting. For a given proof π and random coins R, we denote the corresponding view of the proof by $Q(R, \pi) = (\pi_{i_1(R)}, \pi_{i_2(R)}, \ldots, \pi_{i_q(R)})$.[1]

We are now ready to give the description of the graph $G = (V, E)$. The graph G will have $|V| = 2^{r+q}$ vertices, distributed over 2^r layers, each layer consisting of 2^q vertices. The 2^r layers correspond to the 2^r different random coins, while the 2^q vertices within each layer correspond to the possible 2^q views. Thus,

$$V(G) = \{(R, \mathrm{view}) | R \in \{0,1\}^r, \mathrm{view} \in \{0,1\}^q\}.$$

Two vertices (R, view) and (R', view') are connected by an edge if both the views are accepting and furthermore they do not contradict each other. In other words, there exists a proof π such that (i) $\mathrm{view} = Q(R, \pi)$, (ii) $\mathrm{view}' = Q(R', \pi)$ and (iii) $\mathsf{Ver}^\pi[H; R] = \mathsf{Ver}^\pi[H; R'] = \mathsf{acc}$. This completes the description of the graph $G = (V, E)$.

We now discuss the size of the largest clique in the two cases, depending on whether $H \in$ 3-COLOR or $H \notin$ 3-COLOR.

Completeness: If $H \in$ 3-COLOR, then there exists a proof π such that $\Pr_R[\mathsf{Ver}^\pi[H; R] = \mathsf{acc}] \geq c$.

Consider the following set of vertices.

$$C_\pi = \{(R, Q(R, \pi)) | \mathsf{Ver}^R[H; R] = \mathsf{acc}\}.$$

Clearly the vertices given by C_π form a clique since they correspond the accepting views from the same proof π. We have, $|C_\pi| \geq c\ 2^r$. Thus, in this case, we have $CLIQUE(G) \geq c2^r$.

[1] The above description assumes the verifier is non-adaptive. We could do a similar argument if the verifier were adaptive. The views in this case as in the non-adaptive case are a sequence of q bits in the proof. But the positions in the proof these views correspond to, will be different. For instance, in this case $Q(R, \pi)$ is defined as $Q(R, \pi) = (\pi_{i_1(R)}, \pi_{i_2(R, \pi_{i_1(R)})}, \pi_{i_3(R, \pi_{i_1(R)})}, \pi_{i_2(R, \pi_{i_1(R)})}, \ldots,)$. However, for simplicity we will assume the verifier is non-adaptive.

Soundness: if $H \notin$ 3-COLOR, then for all proofs π, $\Pr_R[\mathsf{Ver}^R[H;R] = \mathsf{acc}] \leq s$. In this case, we will show that $CLIQUE(G) \leq s2^r$. Suppose otherwise, then there exists a set of C vertices in G of size $s2^r$ that form a clique. Since edge exist only between non-contradicting accepting views, there exists a proof π such that for all $(R, \text{view}) \in C$, we have view $= Q(R,\pi)$ and $\mathsf{Ver}^\pi[H;R] = \mathsf{acc}$. But then, we have $\Pr_R[\mathsf{Ver}^\pi[H;R] = \mathsf{acc}] \geq s$, contradicting that $H \notin$ 3-COLOR.

Hence, the reduction $H \mapsto \langle G, c2^r\rangle$ is a reduction from 3-COLOR to gap-CLIQUE$_{s/c}$. Furthermore, this reduction can be performed in time at most linear in the size of the graph G (i.e., 2^{r+q}). This completes the proof of the lemma □

2.3.1 Improving the inapproximability factor

We observe that the inapproximability factor in the above reduction only depends on the ratio s/c. However, by a simple sequential repetition of the verifier improves this factor to any constant $\alpha > 0$ as shown in the following proposition.

Proposition 2.11. *For all $k > 0$, $PCP_{c,s}[r,q] \subseteq PCP_{c^k,s^k}[kr,kq]$*

Proof. Sequentially repeat the actions of the verifier of "$PCP_{c,s}[r,q]$" k-times and accept only if all the k views are accepting. □

Combining this proposition with any constant k with the reduction, we have the improved hardness result.

Corollary 2.12. *$\forall \alpha > 0$, gap-CLIQUE$_\alpha$ is NP-hard.*

We can improve the hardness factor further by choosing k to be super-constant. However, then the number of random coins tossed by the verifier $kr = O(k \log n)$ becomes super logarithmic and hence the running time of the reduction 2^{kr+kq} becomes super-polynomial. The problem here is that we are using kr random coins to repeat the verifier k times. We can instead use techniques from derandomization to recycle random coins. In fact, it is known that $r + O(k)$ (as opposed to kr) random coins suffice to repeat a randomized protocol k times achieving the same exponential improvement in error (see exercise 1 for more details).

Lemma 2.13. *For all k, $PCP_{1,s}[r,q] \subseteq PCP_{1,2s^k}[r+O(k),kq]$.*

Combining this with the hardness result, we get

Corollary 2.14. *There exists a $\delta > 0$, gap-CLIQUE$_{n^{-\delta}}$ in NP-hard. In other words, approximating MAX-CLIQUE to a factor better than $n^{-\delta}$ is NP-hard.*

It is known that we can "recycle queries" and improve the inapproximability factor to $n^{-(1-\epsilon)}$ for any ϵ (albeit under randomized reductions) [Hås99]. We will not cover these results in this course. Note this is almost optimal, since outputting a single vertex gives a $1/n$ approximation algorithm for MAX-CLIQUE.

References

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. *Proof verification and the hardness of approximation problems.* J. ACM, 45(3):501–555, May 1998. (Preliminary Version in *33rd FOCS*, 1992). doi:10.1145/278298.278306.

[AS98] Sanjeev Arora and Shmuel Safra. *Probabilistic checking of proofs: A new characterization of NP*. J. ACM, 45(1):70–122, January 1998. (Preliminary Version in *33rd FOCS*, 1992). doi:10.1145/273865.273901.

[BFL91] László Babai, Lance Fortnow, and Carsten Lund. *Non-deterministic exponential time has two-prover interactive protocols*. Computational Complexity, 1(1):3–40, 1991. (Preliminary Version in *31st FOCS*, 1990). doi:10.1007/BF01200056.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. *Checking computations in polylogarithmic time*. In *Proc. 23rd ACM Symp. on Theory of Computing (STOC)*, pages 21–31. New Orleans, Louisiana, 6–8 May 1991. doi:10.1145/103418.103428.

[BM88] László Babai and Shlomo Moran. *Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes*. J. Computer and System Sciences, 36(2):254–276, April 1988. doi:10.1016/0022-0000(88)90028-1.

[BS05] Eli Ben-Sasson and Madhu Sudan. *Simple PCPs with poly-log rate and query complexity*. In *Proc. 37th ACM Symp. on Theory of Computing (STOC)*, pages 266–275. Baltimore, Maryland, 21–24 May 2005. doi:10.1145/1060590.1060631.

[Din07] Irit Dinur. *The PCP theorem by gap amplification*. J. ACM, 54(3):12, 2007. (Preliminary Version in *38th STOC*, 2006). doi:10.1145/1236457.1236459.

[FGL+96] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. *Interactive proofs and the hardness of approximating cliques*. J. ACM, 43(2):268–292, March 1996. (Preliminary version in *32nd FOCS*, 1991). doi:10.1145/226643.226652.

[FRS94] Lance Fortnow, John Rompel, and Michael Sipser. *On the power of multi-prover interactive protocols*. Theoretical Comp. Science, 134(2):545–557, 21 November 1994. (Preliminary Version in *3rd IEEE Symp. on Structural Complexity*, 1988). doi:10.1016/0304-3975(94)90251-8.

[GLST98] Venkatesan Guruswami, Daniel Lewin, Madhu Sudan, and Luca Trevisan. *A tight characterization of NP with 3-query PCPs*. In *Proc. 39th IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 18–27. Palo Alto, California, 8–11 November 1998. doi:10.1109/SFCS.1998.743424.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. *The knowledge complexity of interactive proof systems*. SIAM J. Computing, 18(1):186–208, February 1989. (Preliminary Version in *17th STOC*, 1985). doi:10.1137/0218012.

[Hås99] Johan Håstad. *Clique is hard to approximate within $n^{1-\epsilon}$*. Acta Mathematica, 182(1):105–142, 1999. (Preliminary Version in *28th STOC*, 1996 and *37th FOCS*, 1997). doi:10.1007/BF02392825.

[Hås01] ———. *Some optimal inapproximability results*. J. ACM, 48(4):798–859, July 2001. (Preliminary Version in *29th STOC*, 1997). doi:10.1145/502090.502098.

[Raz98] Ran Raz. *A parallel repetition theorem*. SIAM J. Computing, 27(3):763–803, June 1998. (Preliminary Version in *27th STOC*, 1995). doi:10.1137/S0097539795280895.