CS 880: Advanced Complexity Theory	3/22/06
Lecture 19: The PCP Theorem	
Instructor: Dieter van Melkebeek	Scribe: Siddharth Barman, Anand Sinha

This lecture deals with the PCP Theorem, its connection with hardness of approximation and provides some preliminaries required for the proof of the PCP Theorem using expanders. In general PCP or Probabilistically Checkable Proofs are a particular type of proof system. The characteristic of this proof system is that we need to check only a few bits to be convinced (to an extent) about its correctness.

1 PCP Theorem

Formally

Definition 1. *A PCP for a language L is a probabilistic polytime oracle TM st*
$x \in L \Rightarrow (\exists\ \Pi)\ \Pr[V^{\Pi}(x)\ \mathit{Accepts}] \geq c$
$x \notin L \Rightarrow (\forall\ \Pi)\ \Pr[V^{\Pi}(x)\ \mathit{Accepts}] \leq s$

The parameters for the proof system are completeness c which is typically 1, soundness s typically set to $\frac{1}{2}$ along with
$r = \#$ of random bits V uses
and $q = \#$ of probes V makes to Π.

The PCP theorem essentially provides an alternate characterization of NP. Below we provide the so called weak version of the PCP Theorem.

Theorem 1. *There exists a constant $s < 1$ s.t. $\mathsf{NP} = \mathsf{PCP}_{1,s}(O(\log n), 3)$ i.e. class NP can be viewed as a PCP system with $c = 1$, $r = O(\log n)$ and $q = 3$.*

The theorem is profound both in content and implications. Astonishingly it suggests the existence of a proof system using which we can be convinced that a formula is satisfiable just by probing 3 bits of a proof. An alternate statement of the theorem is as follows.

Theorem 2. *There exists a constant $s < 1$ and polytime transformation T that maps $3CNF$ formulas to $3CNF$ formulas s.t.*
$\phi \in SAT \Rightarrow T(\phi) \in SAT$
$\phi \notin SAT \Rightarrow$ No assignment to variables of $T(\phi)$ can satisfy a fraction $> s$ of the clauses of $T(\phi)$.

To see the equivalence of the above two theorems we do the following, given a language $L \in \mathsf{NP}$ we convert it to a satisfiability formula ϕ. Then transformation T to get $T(\phi)$. Pick a random clause (this requires $O(\log n)$ random bits as the number of clauses might be polynomial in n). Finally we query the 3 variables present in the clause to verify whether the clause is satisfied or not. For the other direction we generate a list of $3CNF$ formulas by cycling over all possible choices for the random bits(giving polynomial number of formulas), each time we have 3 literals hence each of the formula is of constant size. We determine the fraction s of the list that are verified and hence decide whether the language is in PCP or not.

Note that stronger versions of the PCP theorem exist where in s is set close to $\frac{7}{8}$. The probability that a clause of 3 literals is satisfied by a random assignment is $\frac{7}{8}$ hence the bound for s is tight.

2 Hardness of Approximation

PCP Theorem turns out to be crucial in determining bounds on how close one can get to an optimal solution of an NP Hard optimization problem in polynomial time. The intuition lies in the fact that the prover in the PCP case tries to solve an optimization problem. Consider assignments of variables of a boolean formula in an optimization setting. An optimal solution is one that satisfies all the the clauses of the formula and a good approximation is one that satisfies a large fraction of clauses. By PCP Theorem we know that we can convert an unsatisfiable formula ϕ to $T(\phi)$ such that no assignment to variables of $T(\phi)$ can satisfy a fraction $> s$ of the clauses. Now if were able to determine good approximations we could tell whether a $\phi \in SAT$ by running the polytime approximation on $T(\phi)$. Hence unless $\mathsf{P} = \mathsf{NP}$ no such approximation exists. Some hardness results are stated below

	Best known lower bound	Best known approx factor	Assumption
Vertex Cover	2	1.360	$\mathsf{P} \neq \mathsf{NP}$
		2 - ϵ	UGC[1]
Max Cut	$\alpha = .878$	.942S	$\mathsf{P} \neq \mathsf{NP}$
		$\alpha + \epsilon$	UGC
Set Cover	$\ln n$	$\Omega(\log n)$	$\mathsf{P} \neq \mathsf{NP}$
		$(1 - o(n)) \ln n$	$\mathsf{NP} \subsetneq \mathsf{DTIME}(n^{O(\log \log n)})$
Clique	$\frac{\log^2 n}{n}$	$\frac{1}{n^{1-\epsilon}}$	$\mathsf{P} \neq \mathsf{NP}$

3 Gap Amplification

Now we go on to phrase the PCP Theorem in terms of graphs and provide a proof sketch of the PCP Theorem using gap amplification. We may generalize 3SAT to constraint satisfaction problem as a constraint graph problem. Formally a constraint graph instance $G = ((V, E), \Sigma, C)$. Here Σ is the set of alphabets, the values of variables V and E is the set of constraints. $C = \{c_e | e \in E\}$ with $c_e : \Sigma \times \Sigma \to \{0, 1\}$. A natural instance of the constraint graph is 3 colorability.

We use $Unsat(G)$ to denote the min fraction of constraints that need to be violated under any assignment. Hence the PCP Theorem using CG (Constraint Graph) may be stated as:

Theorem 3. *There exists constant $u > 0$ s.t. for any fixed Σ there exists a polytime transformation T mapping CG G to another CG $T(G)$ with the same Σ s.t.*
$Unsat(G) = 0 \Rightarrow Unsat(T(G)) = 0$
$Unsat(G) \neq 0 \Rightarrow Unsat(T(G)) \geq u$

Lemma 1. *The Gap Amplification Lemma asserts the existence of a polytime computable transformation T' with the following property $Unsat(T'(G)) \geq 2min\{Unsat(G), u\}$ and $size(T(G)) = O(size(G))$.*

The CG version of the PCP theorem is a direct consequence of the Gap Amplification Lemma. Essentially we apply the gap amplifier transformation T' logarithmic number of times to get the

[1] UGC: Unique Games Conjecture

desired transformation T. A formal proof of the lemma will be given next time, for now we consider some preliminaries necessary for proving the gap amplification lemma.

4 Preliminaries for Proof

4.1

$\phi(G)$ the edge expansion of a graph G for $S \subseteq V$ is defined as:

$$\phi(G) = \begin{array}{c} min \\ |S| \leq N/2 \end{array} \frac{E(S,\overline{S})}{|S|}$$

Lemma 2. $\phi(G) \geq \frac{d}{2}(1 - \lambda_2)$

Proof. Recall that

$$\lambda_2 = \begin{array}{c} max \\ x \perp u, x \neq 0 \end{array} \frac{|x^T Ax|}{x^T x}$$

Now the numerator $x^T Ax = x^T x - \frac{1}{2d} \sum_{(i,j)\in E'} (x_i - x_j)^2$.
We set probability vector

$$\begin{aligned} x_v &= \frac{1}{|S|} \quad if\ v \in S \\ &= \frac{1}{|\overline{S}|} \quad o/w \end{aligned}$$

So,

$$x_v^T Ax_v = \left(\frac{1}{|S|} + \frac{1}{|\overline{S}|} - \frac{1}{2d} 2E(S,\overline{S}) \left(\frac{1}{|S|} + \frac{1}{|\overline{S}|} \right)^2 \right)$$

or

$$\lambda_2 x^T x \geq \left(\frac{1}{|S|} + \frac{1}{|\overline{S}|} - \frac{1}{2d} 2E(S,\overline{S}) \left(\frac{1}{|S|} + \frac{1}{|\overline{S}|} \right)^2 \right)$$

Working out the inequality we get

$$E(S,\overline{S}) \geq \frac{(1-\lambda_2)d|S||\overline{S}|}{|S| + |\overline{S}|}$$

Since $\frac{|\overline{S}|}{|S|+|\overline{S}|} \geq \frac{1}{2}$ the lemma holds. $\square$

4.2

Finally we state a lemma about the hitting property of random walks along edges (the proof is based on expander mixing lemma and is left as an exercise).

Lemma 3. *For any set $B \subseteq E$ and $i \geq 0$ the* $\Pr[(i+1)th$ *step is along edge in* $B \mid$ *first step is along random edge in* $B\ ] \leq \frac{|B|}{|E|} + \lambda_2^i$.