Lecture 15

The PCP Theorem

May 25, 2004
Lecturer: Paul Beame
Notes: Ashish Sabharwal

At the end of last class we had nearly finished the proof of the following two theorems.

Theorem 15.1 (Babai-Fortnow-Levin-Szegedy). $\mathsf{NP} \subseteq \mathsf{PCP}(\mathrm{polylog}, \mathrm{polylog})$*; more precisely* $\mathsf{NP} = \mathsf{PCP}(\log n, \log^3 n)$.

Theorem 15.2 (Feige-Goldwasser-Lovasz-Safra-Szegedy). $\mathsf{NP} \subseteq \mathsf{PCP}(\log n \log\log n, \log n \log\log n)$.

Proof. Proof continued To review, in the last lecture, we first described PCP proofs that involved constructing a multilinear extension $\bar{a}$ of an assignment a and checking the arithmetized clauses $CC(w, a)$ for $w \in \{0,1\}^{3\log n+3} = \{0,1\}^{\ell}$ where each arithmetized clause had maximum degree 2 and total degree at most 4. We wanted to check that all these evaluated to 0. So, we used a Reed-Muller code that involved $r_1, r_2, \ldots, r_\ell \in_R I \subseteq \mathbb{F}$ and checked using the sum-check protocol that the polynomial sum $E_R(\bar{a}) = \sum_{w\in\{0,1\}^\ell} CC(w,a) \prod_{i,\, w_i=1} r_i = 0$. The sum-check protocol required a table of size $|I|^\ell \log|\mathbb{F}|$, the number of random bits used was $2\ell \log|I|$ and the number of queries was $\ell \log|\mathbb{F}|$. For the sum-check protocol to have a small failure probability we needed that $|I|$ is at least a constant factor larger than (#vars) $\cdot$ (maxdegree) $= 2\ell$.

In order to apply this test we need to ensure that the table $\bar{a}$ really is close to a multilinear function. To do this we used the aligned line test described in Lecture 13. From Corollary 13.3, for $|I| = O((\text{\#vars})^2 \cdot (\text{maxdegree})^2/\delta^2)$, the number of trials needed to show distance at most some small constant δ was $O((\#vars) \cdot (\text{maxdegree})/\delta) = O((\#vars) \cdot (\text{maxdegree}))$. Each trial required $O(\text{maxdegree})$ queries. The total number of queries is $O((\text{\#vars})(\text{maxdegree}^2)$. Plussing in the number of variables $\ell = \Theta(\log n)$ and the the maximum degree we test for which is 1 in the case of multilinearity we conclude that for $|I| = \Theta(\log^2 n)$, the total number of queries used in the multilinearity test is $O(\log n)$ each of which is an element of $\mathbb{F}$ which will require $O(\log|\mathbb{F}|) = O(\log\log n)$ bits. This is fine for what we wanted.

However, by Corollary 13.3, each trial uses (#vars) $\log|I|$ random bits so the total number of random bits used is $\Omega((\text{\#vars})^2(\text{maxdegree})\log|I|)$, which is $\Omega(\log^2 n \log\log n)$. This was more than our target of $O(\log n \log\log n)$ random bits. To fix this we can use pairwise independence in place of complete independence between the trials. We previously discussed this idea, due to Chor and Goldreich, in theory seminar in Winter Quarter.

Pairwise Independent Trials Suppose we have a test that uses r random bits per trial. Using pairwise independent hash functions $h_{a,b}(x) = ax + b$ over $\mathbb{F}_{2^r}$, we can do k' trials by choosing $a, b \in_R \mathbb{F}_{2^r}$ and

using $h_{a,b}(1), h_{a,b}(2), \ldots, h_{a,b}(k')$ for $k' < 2^r$ as pairwise independent random strings. This uses only $2r$ random bits overall. By Chebyshev's inequality, with high probability, the number of successful trials is close to the expected number when the trials are completely independent, for large enough k'.

In our case, $r =$ (#vars) $\log |I|$ random bits per trial are used and $k' = c$(#vars) $\cdot$ (maxdegree) trials is still sufficient with the additional errors introduced through the Chebyshev inequality. Thus a total of $O(\log n \log\log n)$ random bits suffice for the aligned line test of multilinearity. This is enough to yield the theorem that $\mathsf{NP} \subseteq \mathsf{PCP}(\log n \log\log n, \log n \log\log n)$.

In order to reduce the proof size to polynomial and the number of random bits to $O(\log n)$ toward the end of the last lecture we replaced the two element set $\{0,1\}$ with set H with $|H| = h = \log n$. We could now encode n variables with $\log n / \log\log n$ variables over the set $\{1, 2, \ldots, h\}$ by a simple change of basis. However, by this change, the maximum degree $k = 2$ in the sum-check protocol goes up to $O(\log n)$ since the assignment A is degree $h - 1$ and is no longer multilinear. Similarly, in the max-degree $h - 1$ test of the assignment A, the (maxdegree)^2 term which previously was constant now becomes significant and thus the total number of queries q grows to $O(\log^3 n)$ bits. The size of I also needs to grow to $\Theta(\log^4 n)$ to compensate for the growth in the max degree. However, using the pairwise independent trials the number of random bits from both the max-degree-k tests and the sum-check protocol are still $O($(#vars) $\log |I|)$ random bits which is now only $O(\log n)$ bits. Thus $\mathsf{NP} = \mathsf{PCP}(\log n, \log^3 n)$ follows. □

15.1 The PCP Theorem

The rest of this lecture will be devoted to the proof and implications of the following result:

Theorem 15.3 (PCP Theorem). $\mathsf{NP} = \mathsf{PCP}(\log n, 1)$

15.1.1 Implications on Hardness of Approximation

Before going into the proof of the PCP theorem, we give one example of what it implies for approximation problems. Let MAX3SAT be the problem of finding an assignment to a given 3CNF formula F that maximizes the number of clauses of F satisfied. An *approximation algorithm* for MAX3SAT with approximation factor γ finds an assignment that satisfies at least OPT/γ clauses, where OPT is the number of clauses satisfied by an optimal assignment.

MAX3SAT is a complete problem in the class MAXSNP of NP optimization problems introduced by Papadimitriou and Yannakakis. Each problem in this class has a constant factor polynomial time approximation algorithm but it is not known whether or not these approximation factors can be made arbitrarily close to 1. The following corollary of the PCP Theorem shows that this is unlikely in general.

Corollary 15.4. *There exists an $\epsilon > 0$ such that if there is a polytime $(1+\epsilon)$-approximation for* MAX3SAT *then* $\mathsf{P} = \mathsf{NP}$.

Proof. We will convert SAT into a 'gap' problem and show that if there is a good enough approximation to MAX3SAT, then SAT can be solved in polynomial time. The PCP theorem yields a polynomial time verifier $V^?_{\mathrm{SAT}}$ with the following behavior. Given a formula ψ and an assignment a, consider the polysize PCP proof $E(a)$ that a satisfies ψ. $V^?_{\mathrm{SAT}}$ looks at $E(a)$, uses $O(\log n)$ random bits, and makes $q = O(1)$ queries based on those bits. If $\psi \in \mathrm{SAT}$, then $V^?_{\mathrm{SAT}}$ accepts. If $\psi \notin \mathrm{SAT}$, then V_{SAT} accepts with probability at most $1/2$.

$V^?_{\text{SAT}}$ has $2^{O(\log n)}$ = polynomial number of random choices. Create one test circuit for each such choice r, $Test_r : \{0,1\}^q \to \{0,1\}$. The inputs to each of these test circuits are (different) $q = O(1)$ bits of $E(a)$. Convert each $Test_r$ into a circuit of size $O(2^q/q)$; this can be done because of Lupanov's bound on circuit size stated in Lecture 4. (A trivial bound of $q2^q$ is also sufficient since these are all constant size.) Now convert each circuit into a 3CNF by adding extra variables. This is again of size $O(2^q/q)$. Call the conjunction of all these 3CNF's ψ'. As described above, ψ' can be constructed from ψ in polynomial time.

If $\psi \in$ SAT, then $\psi' \in$ SAT. If $\psi \notin$ SAT, then for any assignment a, at least half the tests $Test_r$ are not satisfied and thus have at least one unsatisfied clause in each, implying that a total of at least a $q/(2 \cdot 2^q) = \Omega(q/2^q)$ fraction of clauses of ψ' are unsatisfied. Since $q = O(1)$, this is a constant fraction and we have a gap problem for CNF satisfiability. If MAX3SAT can be approximated within a constant factor, then this gap problem can be solved exactly, proving the Corollary. □

15.1.2 Outline of the PCP Theorem Proof

As shown in Lecture 12, it will be sufficient to prove that $\mathsf{NP} \subseteq \mathsf{PCP}(\log n, 1)$ because any proof on the right hand side can be trivially converted into an NP proof by enumerating the underlying tests for all possible $2^{O(\log n)} = n^{O(1)}$ random choices of the PCP verifier.

Variants of the Low Degree Test

We used the aligned line test for max-degree k in the PCP constructions above. Even with an improved analysis due to Arora and Safra, this test is not efficient enough for the PCP Theorem.

In the polynomial-size PCP construction above, the polynomials had max-degree $k = \log n$. Their total degree was not much larger, since the number of variables $\ell = \log n / \log\log n$ and thus their total degree $d = \log^2 n / \log\log n$. Let $P_{tot}(\ell, d)$ be the set of all total degree d polynomials in ℓ variables.

In the improvements of the above PCP constructions that yield the PCP theorem, an efficient total-degree test instead of an max-degree test is used. Since the original PCP Theorem there have been a number of total-degree tests proposed and the original analysis has been improved. These tests all work for the construction and provide different analysis complexity, exact PCP proof size and other parameter variations.

The original test, which is also easy to describe, is the Affine Line Test due to Arora, Lund, Motwani, Sudan, and Szegedy.

The Affine Line Test: To test a function $f : \mathbb{F}^\ell$

1. Choose $x, y \in_R \mathbb{F}^\ell$
2. Query the $d+1$ coefficients of $f_{x,y}(t)$ where we are supposed to have $f_{x,y}(t) = f(x + yt)$
3. Check that $f_{x,y}(t) = f(x + yt)$ holds at a random point $t \in_R \mathbb{F}$.

Note that $(x + yt)$ for different t's are uniformly spaced points along a line of slope y with x as the starting point. We state the following theorem without proof.

Theorem 15.5. *The affine line test has the following properties:*

1. *If f has total degree at most d, then the test always accepts.*

2. *There is some $\delta_0 > 0$ such that if the test accepts with probability at least $1 - \delta$ for some $\delta < \delta_0$, then f is within a 2δ fraction of a total degree d polynomial, i.e. , $d(f, P_{tot}(\ell, d)) \leq 2\delta$.*

Note that this theorem is much more efficient both in number of queries and random bits than the aligned line test. We now give a sketch of rest of the proof of the PCP theorem. Many of the details will be omitted for lack of time.

15.1.3 Three New Ideas

There are three additional key ideas that are used in the proof of the PCP theorem.

Self-correction. The original protocol in the last lecture examined $\bar{a}$ in only three random places. In general the proof will need a more flexible way to access the truth assignment $\bar{a}$ or other polynomials in the proof.

Suppose we know that $d(\bar{a}, P_{total}(\ell, d)) \leq \delta$. Let $\hat{a}$ be the closest point in $P_{total}(n, d)$. We would like to evaluate $\hat{a}$ instead of $\bar{a}$.

Claim 3. We can evaluate $\hat{a}$ at an arbitrary place and be correct with probability at least $1 - (d+1)\delta$.

Proof idea. To evaluate $\hat{a}(y)$, choose a random y and compute $\bar{a}(x+y), \bar{a}(x+2y), \ldots, \bar{a}(x+(d+1)y)$. Compute degree d polynomial $p(i) = \bar{a}(x + iy)$ by interpolation. Evaluate $\hat{a}(x) = p(0)$. The randomness used here is in selecting $y \in_R \mathbb{F}^\ell$. The number of random bits needed is therefore the same as before. The number of queries is $d + 1$. □

A Completely Different PCP.

Lemma 15.6 (Arora-Lund-Motwani-Sudan-Szegedy). $\mathsf{NP} \subseteq \mathsf{PCP}(n^2, 1)$

We will describe the main ideas behind this result in the next section.

Composition. [Arora-Safra] We know $\mathsf{NP} = \mathsf{PCP}(\log n, \text{polylog } n)$. Start with such a PCP verifier V. Given a PCP proof, V uses $O(\log n)$ random bits and checks $\log^c n$ places of the proof, for some constant $c \geq 0$. Here is where the idea of composition comes in – instead of checking all $n' = \log^c n$ bits of the proof, view this test as an NP-style check on n' bits. This NP-style test itself can be replaced by an "inner" PCP verifier with parameters $(n'^2, 1)$ using the completely different PCP above. The new PCP proof now contains the original "outer verifier" of type $\mathsf{PCP}(\log n, \text{polylog } n)$ and for each test on n' bits, an "inner verifier" of type $\mathsf{PCP}(n'^2, 1)$ that makes sure the test is satisfied. In fact, we will need to apply composition more than once.

For this composition to make sense, we also need to add a "consistency check" to make sure all assignments in the inner verifiers are consistent with a single outer verifier table. This part of the construction is quite hairy and uses very specific properties of the proof tables themselves. We will skip the details.

15.1.4 Outline of the Proof

Assume for now that all three of these new ideas work out. We get

$$\begin{aligned} \mathsf{NP} &\subseteq \mathsf{PCP}(\underbrace{\log n}_{\text{outer}} + \underbrace{\log^{2c} n}_{\text{inner}}, 1) \\ &= \mathsf{PCP}(\text{polylog } n, 1) \end{aligned}$$

Now do a composition again with this new verifier as the inner verifier. We get

$$\begin{aligned} \mathsf{NP} &\subseteq \mathsf{PCP}(\underbrace{\log n}_{\text{outer}} + \underbrace{\text{poly}(\log\log n)}_{\text{inner}}, 1) \\ &= \mathsf{PCP}(\log n, 1) \end{aligned}$$

15.1.5 Idea Behind $\mathsf{NP} \subseteq \mathsf{PCP}(n^2, 1)$

Let QUADRATIC-SAT be the following problem: given a family of polynomial equations of total degree at most 2 over $\mathbb{F}_2$, are they simultaneously satisfiable?

Theorem 15.7. QUADRATIC-SAT *is* NP*-complete.*

Proof idea. One can simulate CIRCUIT-SAT using this problem. The general idea is to use the quadratic equations to define the gates variables in the same way that such variables are used in conversion of CIRCUIT-SAT into 3SAT. For instance, an AND gate with inputs y_i and y_j, and output y_k translates into the total degree 2 equation $y_k = y_i \cdot y_j$ over $\mathbb{F}_2$. □

We will prove that QUADRATIC-SAT is in $\mathsf{PCP}(n^2, 1)$. Suppose we have an assignment a and polynomials $P_1, P_2, \ldots, P_m$ of total degree at most 2. The PCP verifier will work as follows.

1. Choose $r_1, r_2, \ldots, r_m \in \mathbb{F}_2$.
2. Check assignment a on the total degree at most 2 polynomial $P \equiv r_1P_1 + r_2P_2 + \ldots + r_mP_m$, *i.e.* , test whether $P(a) = 0$.

By our standard argument, if there is some i such that $P_i(a) \neq 0$ then the probability that $P(a) = 0$ is $1/2$.

The verifier will know the coefficients of P but they depend on the random choices of the r_i that are not know in advance so the proof table will need to give values for all possible quadratic polynomials. The proof table that the verifier will access to do this calculation corresponds to expressing a total degree 2 polynomial as:

$$P = b_0 + \sum_{i=1}^{n} b_i y_i + \sum_{i,j=1}^{n} c_{ij} y_i y_j$$

The PCP proof will include a table of $\sum_{i=1}^{n} b_i a_i$ for all possible $b \in \mathbb{F}_2^n$ as well as a table of $\sum_{i,j=1}^{n} c_{ij} a_i a_j$ for all possible $c \in \mathbb{F}_2^n$. The former of these is the well-known Hadamard code from coding theory, while the latter is what is called the Quadratic code.

To check that this is correct, the verifier needs to do the following three things:

Check linearity of the tables in b and c, resp. Let $f(b)$ denote the result of querying the first table on b. To check linearity, the verifier checks $f(b \oplus b') = f(b) \oplus f(b')$ for $b, b' \in_R \mathbb{F}_2^n$. This requires 3 queries and we state without proof that if the check succeeds with probability at least $1 - \delta$ then the function differs from a linear function in at most a δ fraction of entries. Linearity of the second table in c is verified similarly.

Self-correction. Because the specific query b needed for P might be at a place where the table is incorrect, so to compute $f(b)$, choose a $b' \in_R \mathbb{F}_2^n$ and instead compute $f(b) \leftarrow f(b \oplus b') \oplus f(b')$.

Consistency check between the two tables. Although the individual b and c tables may be individually linear there might be no relationship between them. For any $y \in \mathbb{F}^n$ and $b, b' \in_R \mathbb{F}_2^n$ observe that $(\sum_i b_i y_i)(\sum_j b'_j y_j) = \sum_{i,j} b_i b'_j y_i y_j$. Thus for consistency we need that $c_{ij} = b_i b'_j$. In particular we need to check that that $(\sum_i b_i a_i)(\sum_j b'_j a_j) = \sum_{i,j} b_i b'_j a_i a_j$. The first two summations can be evaluated directly from the first table, while the last summation can be evaluated using the second table with $c_{ij} = b_i b'_j$. Using self-correction for each of the three summation evaluations, this takes 6 queries.

The verifier uses $O(n^2)$ random bits overall to perform these checks and makes only a constant number of queries. This proves that $\mathsf{NP} \subseteq \mathsf{PCP}(n^2, 1)$.