CS286.2 Lecture 2: Equivalence of two statements of PCP, and a toy theorem

Scribe: Fernando Granha

In this second lecture, we show how to go from the CSP version of the PCP Theorem to its Game variant. Moreover, we state a simplified PCP Theorem whose proof nonetheless uses some of the tools and ideas from the original one. We begin the proof of the simplified PCP.

(PCP, CSP variant) $\implies$ (PCP, games variant)

The next Lemma shows how to go from a CSP promised to have a constant gap in $\omega(\varphi_x)$ to a game G_x which also has a constant gap in $\omega(G_x)$.

Lemma 1. *Given a (m,q)-CSP instance φ_x on n variables promised by the (PCP, CSP variant) Theorem, it is possible to construct a Game G_x such that:*

(i) $\omega(\varphi_x) = 1 \implies \omega(G_x) = 1$, *and*

(ii) $\omega(\varphi_x) \leq \frac{1}{2} \implies \omega(G_x) \leq 1 - \frac{1}{10q^2}$.

As shown in the exercises, by repeating the game G_x in parallel sufficiently many times with independent sets of players it is possible to reduce the game value from $1 - \frac{1}{10q^2}$ to $1/2$ in case (ii) (while preserving value 1 in case (i)), thereby completing this step of the equivalence.

Proof. Before we start the proof, to make the notation precise, we definite a function $f : \{1, \ldots, m\} \times \{1, \ldots, q\} \to \{1, \ldots, n\}$, that takes the index of a constraint index and the index of a variable appearing in that constraint to the index of this variable in $\{1, \ldots, n\}$. Consider the following game with two players P_1 and P_2.

```
Choose uniformly at random a constraint C_j(z_{f(j,1)}, ..., z_{f(j,q)}) for j ∈ {1, ..., m};
Choose uniformly at random a variable i in {1, ..., q} ;
Ask P_1 for an assignment to the variables in C_j;
Denote by a^1_{f(j,1)}, ..., a^1_{f(j,q)} the answers received for z_{f(j,1)}, ..., z_{f(j,q)}, respectively;
Ask P_2 for an assignment to z_{f(j,i)} ;
Denote by a^2_{f(j,i)} the answer received ;
Reject if a^1_{f(j,1)}, ..., a^1_{f(j,q)} do not satisfy C_j ;
Accept iff a^2_{f(j,i)} = a^1_{f(j,i)}.
```

Algorithm 1: Referee in G_x

When $\omega(\varphi_x) = 1$, there is an assignment $(a_i)_{i=1,\ldots,n}$ to the variables $z_1, \ldots, z_n$ that satisfies all constraints. Therefore, if both players answer according to it, it is clear that $\omega(G_x) = 1$ concluding item (i).

To show item (ii), we show the contrapositive: $\omega(G_x) > 1 - \frac{1}{10q^2} \implies \omega(\varphi_x) > \frac{1}{2}$. In words, if the value of the game is sufficiently large, then there is an assignment to $z_1, \ldots, z_n$ that satisfies more than half of the constraints. Recall that without loss of generality we may assume that the strategies of the players are deterministic. A strategy for player 2 is a fixed assignment to all variables of φ_x, that we denote by $a^2 = (a^2_i)_{i=1,\ldots,n}$. Given the assumption that $\omega(G_x) > 1 - \frac{1}{10q^2}$ we claim that this assignment satisfies more than half of the constraints. We bound the probability $\Pr[a^2 \text{ satisfies } C_j]$ from below as follows:

$$\Pr[a^2 \text{ satisfies } C_j] \geq \Pr[P_1\text{'s strategy satisfies } C_j \wedge a^2_{f(j,1)} = a^1_{f(j,1)} \cdots \wedge a^2_{f(j,q)} = a^1_{f(j,q)}].$$

By negating the probability in the rhs and using the union bound, we have

$$\Pr[a^2 \text{ satisfies } C_j] \geq 1 - \Pr[P_1's \text{ strategy does not satisfy } C_j] - \Pr[a^2_{f(j,1)} \neq a^1_{f(j,1)}] - \cdots - \Pr[a^2_{f(j,q)} \neq a^1_{f(j,q)}].$$

If P_1's strategy does not satisfy C_j, the referee readily rejects. Consequently, the probability of C_j not being satisfied it at most $1 - \omega(G_x)$. Each time the referee detects a disagreement between $a^2_{f(j,i)}$ and $a^1_{f(j,i)}$ for i in $\{1, \ldots, q\}$ it rejects. The probability that any index $i \in \{1, \ldots, q\}$ is chosen as the second player's question is exactly $1/q$. Therefore for any fixed i, over the choice of a random j, $\Pr[a^2_{f(j,i)} \neq a^1_{f(j,i)}] \leq q(1 - \omega(G_x))$. These observations result in the bound

$$\Pr[a^2 \text{ satisfies } C_j] \geq 1 - (1 - \omega(G_x)) - q(1 - \omega(G_x)) - \cdots - q(1 - \omega(G_x)).$$

Finally, using the hypothesis that $\omega(G_x) > 1 - \frac{1}{10q^2}$, we can conclude that

$$\Pr[a^2 \text{ satisfies } C_j] \geq 1 - \frac{1}{10q^2} - \frac{1}{10q} - \cdots - \frac{1}{10q} \geq 1 - \frac{2}{10} > \frac{1}{2}.$$

□

A "toy" version of the PCP Theorem

The original PCP Theorem in its proof-checking version demonstrates that for any $L \in$ NP there exists a verifier that uses only $O(\log(n))$ random bits, queries only a constant number of positions in the proof, and correctly answers the question $x \in L$? with constant probability. A simpler version only requires the number of random bits to be polynomial in the input size:

Theorem 2. NP $\subseteq$ PCP$(r = O(poly(n)), q = O(1))$.

This version has an exponential blowup in the maximal proof size that is $O(2^{poly(n)})$ compared to $O(poly(n))$ from the original PCP Theorem. Despite being a weaker result, it will allow us to demonstrate tools and ideas used in the original version.

In order to prove Theorem 2, we use the NP-complete problem "Quadratic Equations" (**QUADEQ**) that is defined next.

Definition 3. *(**QUADEQ**) An instance φ of **QUADEQ** is given by m constraints C_j over n boolean variables x_i of the form:*

$$C_j : \sum_i \alpha_i^{(j)} x_i + \sum_{i,k} \beta_{i,k}^{(j)} x_i x_k \equiv \gamma^{(j)} \mod 2,$$

or equivalently

$$\alpha^{(j)} \cdot x + \beta^{(j)} \cdot (x \otimes x) \equiv \gamma^{(j)} \mod 2,$$

where

$$x = (x_i)_{i=1,\ldots,n} \in \{0,1\}^n,$$
$$\alpha^{(j)} = (\alpha_i^{(j)})_{i=1,\ldots,n} \in \{0,1\}^n,$$
$$\beta^{(j)} = (\beta_{ik}^{(j)})_{i,k=1,\ldots,n} \in \{0,1\}^{n^2} \textit{ and}$$
$$\gamma^{(j)} \in \{0,1\}.$$

*The instance φ belongs to **QUADEQ** if and only if there is an assignment x that satisfies all constraints.*

The following is an example of a **QUADEQ** instance.

$$\begin{cases} C_1: & x_1 + x_2 + x_4x_5 + x_2x_7 \equiv 1 \mod 2 \\ C_2: & x_7 + x_1x_2 \equiv 0 \mod 2 \\ \vdots & \\ C_m: & x_9 + x_5x_6 \equiv 1 \mod 2 \end{cases} \tag{1}$$

The goal is to describe a PCP verifier for **QUADEQ** and as we advance some tools are established. The first such tool is a test that fails with probability $\frac{1}{2}$ if a **QUADEQ** instance φ is infeasible, and always accepts otherwise.

Given coefficients $a = (a_i)_{i=1,\ldots,m} \in \{0,1\}$ chosen independently and uniformly at random, form an equation by combining the constraints of φ as follows:

$$\mathrm{E} = \mathrm{E}(a) : \sum_j a_j(\alpha^{(j)} \cdot x + \beta^{(j)} \cdot (x \otimes x)) = \sum_j a_j \gamma^{(j)}.$$

Claim 4. *For a uniformly random choice of the coefficients a, it holds:*

(i) If x satisfies all constraints, then x satisfies $\mathrm{E}(a)$,

(ii) If x does not satisfy all constraints, $\Pr_a[x \text{ satisfies } E(a)] \leq \frac{1}{2}$.

Proof. Item (i) is clear, as any assignment that satisfies all equations individually must also satisfy the sum.

For item (ii), we introduce the error vector given by

$$e = \begin{pmatrix} \alpha^{(1)} \cdot x + \beta^{(1)} \cdot (x \otimes x) - \gamma^{(1)} \\ \vdots \\ \alpha^{(m)} \cdot x + \beta^{(m)} \cdot (x \otimes x) - \gamma^{(m)} \end{pmatrix} \tag{2}$$

Since not all the constraints of φ are satisfiable, the vector e has at least one not zero component. Note that the inner product of the random vector a with the error vector e checks the parity of the elements a_i for which $e_i = 1$. As the elements a_i are drawn independently and uniformly at random this parity is 1 with probability exactly $\frac{1}{2}$. Moreover, the probability that x does not satisfy E is $\Pr_a[e \cdot a = 1] \leq \frac{1}{2}$. □

Now, We are ready for our first attempt to solve the simplified PCP Theorem 2. We assume that the verifier has access to a proof $\Pi = (\Pi^1, \Pi^2)$ where $\Pi^1 \in \{0,1\}^{2^n}$ and $\Pi^2 \in \{0,1\}^{2^{n^2}}$. Ideally, we would like to have Π to be composed of

- $(\Pi^1)_\alpha = \alpha \cdot x$, and
- $(\Pi^2)_\beta = \beta x \cdot (\otimes x)$.

for some $x \in \{0,1\}^n$.

In words, the proof Π^1 encodes in each position α the value of the inner product with a fixed x (similarly to Π^2). If $\varphi \in$ **QUADEQ**, the bit string x would be the satisfying assignment.

It is important to note that all combination of the constraints given by any random a will lead to a new value for α and β whose inner product with x is encoded in Π. This is a key point that allows us to use Claim 4. A first attempt at designing a verifier for **QUADEQ** is given below.

1 Choose $a = (a_i)_{i=1,\dots,m} \in \{0,1\}$ uniformly at random ;
2 Compute $\begin{cases} \alpha = \sum_j a_j \alpha^{(j)} \in \{0,1\}^n \\ \beta = \sum_j a_j \beta^{(j)} \in \{0,1\}^{n^2} \\ \gamma = \sum_j a_j \gamma^{(j)} \in \{0,1\} \end{cases}$;
3 Make two queries $(\Pi^1)_\alpha$ and $(\Pi^2)_\beta$;
4 Accept iff $(\Pi^1)_\alpha + (\Pi^2)_\beta = \gamma$;

Algorithm 2: Verifier V for **QUADEQ**

The problem of this verifier is that it expects the proof to be in a particular format. Provided this is the case, it follows from Claim that the verifier V has completeness 1 and soundness at most $\frac{1}{2}$. However, it can not rely on receiving this exact format, or otherwise the system may loose its constant soundness as the proof Π is given by an adversarial prover.

The proofs Π^1 and Π^2 should encode the evaluation of a linear function (the inner product with a fixed x, or $x \otimes x$) over all possible inputs. Fortunately, this is a strong property that we can exploit to ensure that Π is "close" to having the desired format. For this, we devise a linearity test that has oracle access to a function $f : \{0,1\}^n \to \{0,1\}$ and whose goal is to check that f is linear. (By linearity we mean that there is $c \in \{0,1\}^n$ such that $f(\alpha) = c_1\alpha_1 + \cdots + c_n\alpha_n \mod 2 = c \cdot \alpha$ for every α.)

Testing if f is exact linear would require querying its value on all inputs. Nevertheless, the next simple test can enforce that it is "almost" linear.

1 Choose $\alpha, \alpha' \in \{0,1\}^n$ at random;
2 Query $f(\alpha), f(\alpha'), f(\alpha + \alpha')$;
3 Accept iff $f(\alpha + \alpha') = f(\alpha) + f(\alpha')$;

Algorithm 3: BLR Linearity Test

The next theorem makes precise our notion of "almost linear". If the linearity test succeeds with high probability, f agrees with a single linear function on a large fraction of inputs.

Theorem 5 (BLR)**.** *The BLR linearity test satisfies:*

(i) If f is linear, then $\Pr[f$ *passes BLR test*$] = 1$.

(ii) Suppose $\Pr[f$ *passes BLR test*$] \geq 1 - \epsilon$ *for some* $\epsilon > 0$, *then there is a coefficient vector c such that* $f(\alpha) = c \cdot \alpha$ *for* $1 - \epsilon$ *fraction of* $\alpha \in \{0,1\}^n$.