PCPs and HDX - Lecture 5

December 6, 2016

Today I want to talk a little bit about PCPs, i.e. probabilistically checkable proofs. We've already seen that the Hadamard code is locally testable with 3 queries. We have seen that the constraint system containing the linearity tests is an expanding constraint system for this code.

1 Hadamard based PCPs

We will start by constructing an expanding constraint system for slightly more general codes. Suppose you have variables $v_1, \dots, v_n$ and some m linear equations over these variables, where each equation is specified by a coefficient vector $a \in \{0,1\}^n$ and a bit b such that the equation is $\langle a, v\rangle = b$. The set of all solutions to these equations is an affine subspace, $W \subset \{0,1\}^n$, and let us consider the following code:

$$H_W = \{H(w) | w \in W\} \subset \{0,1\}^N,$$

where $N = 2^n$ and $H(w)$ is the Hadamard encoding of $w \in \{0,1\}^n$, i.e.,

$$H(w) = (\langle w, x\rangle)_{x \in \{0,1\}^n}.$$

Is *this code* locally testable? The following is a suggestion for a test: given a string Z:

1. Perform a linearity test.
2. Choose a random linear combination of the m linear equations. Let $a \in \{0,1\}^n$ and $b \in \{0,1\}$ be the resulting equation. Select a random $x \in \{0,1\}^N$ and accept iff $Z(x) = Z(x+a) + b$.

Let us first check the completeness of this test. LSuppose $w \in W$ and let $Z = H(w)$ be an element of the code. Clearly Z passes the test in the first item. Moreover $Z(a) =< w, a >= b$ so it must also pass the second item test.

We can see that the set of strings Z that satisfy every constraint described by the test is exactly our code. We now ask whether this set of local constraints (each constraint checks five entries in the codeword Z) is expanding.

Let Z be some given word. Let $\delta = \mathrm{dist}(Z, H)$, and let $\delta' = \mathrm{dist}(Z, H_W)$.

If $\delta \neq \delta'$ then the closest Hadamard codeword to $Z$is the Hadamard encoding of a word $w' \notin W$. In other words w' fails to satisfy at least one equation. We analyze three cases:

- If $\delta = \delta'$ then the linearity test constraints already guarantee soundness: $\mathrm{rej}(Z) \geq \gamma_{BLR} \cdot \delta'$.
- If $\delta \neq \delta'$ and $\delta > 1/8$ again the linearity test constraints kick in: $\mathrm{rej}(Z) \geq \gamma_{BLR}/8 \geq \gamma_{BLR}/8 \cdot \delta'$.

- If $\delta \neq \delta'$ and $\delta < 1/8$: $\mathrm{rej}(Z) \geq \Pr[\langle w', a\rangle \neq b] \cdot \Pr_x[Z(x)+Z(x+a) = \langle w', a\rangle] \geq 1/2 \cdot 1/2 \geq \delta'/4$.

All in all we see that $\mathrm{rej}(Z) \geq \mathrm{const} \cdot \mathrm{dist}(Z, H_W)$. We have proven,

Theorem 1.1. *Every system of linear equations over n variables spanning the space $W \subset \{0,1\}^n$ gives rise to the code $H_W \subset \{0,1\}^{2n}$ that is locally testable with a constant number of queries.*

What does this mean? It means that there is a proof system for checking whether a given system of equations is satisfiable.
Prover: supply the Hadamard encoding of some solution w to the equations.
Verifier: run the above test.
If the proof causes the verifier to accept with high probability, say above 99%, it means that the proof can be decoded into a valid element in W, i.e., a solution to *all* of the equations. If however $W = \phi$, then the verifier is guaranteed to reject with probability higher than 1%.

This scheme has two drawbacks: the first is that the Hadamard encoding has exponential length. The second is that it only allows us to check solutions for systems of linear equations. This second issue is particularly underwhelming since the verifier can check if $W = \phi$ without need of *any* proof at all. However, we shall next see a PCP system for verifying NP hard systems of equations.

2 Quadratic functions encoding

We will now see our third locally testable code, the QF code.
To encode a string $v \in \{0,1\}^n$ in the QF code we first append 0, defining $v_0 = 1$, and then define $u = v \otimes v$ and finally let $QF(v) = H(u)$. I.e.

$$v \Rightarrow (v_0 = 1, v_1, \ldots, v_n) \Rightarrow u = v \otimes v \Rightarrow H(u)$$

Let

$$QF = \{H(u) \mid u = v \otimes v \text{ for some } v \in \{0,1\}^{n+1} \text{ where } v_0 = 1\}.$$

This code is called the "quadratic functions" code because the encoding of a string v has a coordinate $M = (m_{ij})_{ij}$ for every possible quadratic function $\sum_{ij} m_{ij} v_i v_j$ of v. Here is a siz query test for deciding if a given string Z is in this code.

- Perform a linearity test on Z
- Perform a tensor test:
  - Choose $a, b \in \{0,1\}^{n+1}$ such that $a_0 = 1 = b_0$. Define $A, B \in \{0,1\}^{(n+1)^2}$ to be matrices that are zero except for the top row being a in A and b in B.
  - Let $C = a \otimes b$, i.e. $C_{ij} = a_i b_j$.
  - Accept iff $Z(C) = Z(A)Z(B)$.

Theorem 2.1. *The QF code is locally testable with 6 queries and constant soundness. In other words, the above test is an expanding system of 6-ary constraints for this code, with constant expansion.*

Proof. Let $\delta = \mathrm{dist}(Z, QF)$. If the closest Hadamard encoding to Z is also the closest QF encoding, then expansion is promised through the linearity testing constraints, because

$$\mathrm{rej}(Z) \geq \mathrm{rej}_{BLR}(Z) \geq \gamma_{BLR} \cdot \mathrm{dist}(Z, LIN) = \gamma_{BLR} \cdot \mathrm{dist}(Z, QF)$$

So assume there is a string $H(u)$ that is closest to Z and yet $H(u) \notin QF$. Think of u as an $(n+1)$ by $(n+1)$ matrix and let v be the top row of u. Let $R_{ij} = u_{ij} - v_i v_j$. If $u \neq v \otimes v$ then $R \not\equiv 0$. It is easy to check that

$$\Pr_{a,b}[Z(C) \neq Z(A)Z(B)] = \Pr_{a,b}[\sum_{ij} R_{ij} a_i b_j \neq 0] > 1/4$$

Therefore, in this case, the tensor test rejects with probability at least $\frac{1}{4} \geq \frac{1}{4} \cdot \mathrm{dist}(Z, QF)$. $\square$

3 A PCP for NP

3.1 A PCP for checking quadratic equations

Consider as before the system of n variables, and suppose we have m equations over these variables, however this time the equations will be quadratic. For example an equation can be: $v_1 \cdot v_5 + v_6 \cdot v_2 = 1$.

Let us introduce new variables u_{ij} that are supposed to encode products of the original variables, $u_{ij} = v_i v_j$.
Every quadratic equation in the old variables can be written as a linear equation in the new variables when we also add the convention $v_0 = 1$ (this is called the linearized equation). Let W be the space of all solutions u to the linearized equations.

$$QF_W = \{H(u) \mid u \in W \text{ and } u = v \otimes v \text{ for some } v \text{ where } v_0 = 1\}$$

Theorem 3.1. *For every system of quadratic equations, the code QF_W defined above is locally testable with a constant number of queries.*

The proof of this theorem is immediate from the above theorems. We run three tests

- Perform a linearity test
- Perform a random equation test
- Perform a tensor test. Accept only if all tests accepted

3.2 quadratic equations capture all of NP

Let C be a circuit, with input variables $v_1, \ldots, v_n$. It is easy to encode the computation of the circuit as a system of quadratic equations. Assume without loss of generality that each gate in the circuit has fan in two. Introduce one new variable per gate. For each $\wedge$ gate we add the equation $a = bc$ where a is the gate variable and b, c are the input variables to the gate. For each $\vee$ gate we add the equation $a = b + c - bc$. One can check that every assignment to all of the variables that satisfies every single equation, encodes a valid computation of the circuit.

Together with the above theorem, we get our first PCP theorem:

Theorem 3.2. *$NP \subseteq PCP[poly, 1]$*

Notation: The notation on the right $PCP[r, q]$ refers to a PCP verification system where the verifier uses $O(r)$ random bits and makes $O(q)$ queries to the proof. The theorem says that every NP statement can be checked by a probabilistic verifier that uses a polynomial number of random bits and makes constant number of queries to the proof.

Proof. Every NP complete language can be reduced to the problem of circuit satisfiability. So to prove the PCP theorem it is enough to prove it for circuit satisfiability. A verifier for circuit satisfiability will expect the prover to write down the QF encoding of a string that describes the circuit computation. Using the theorem above, that we have yet to prove, this encoding is locally testable. The verifier will run the local test. If the circuit is satisfiable then there is clearly a proof that will satisfy the verifier with probability one. If the circuit is unsatisfiable, the QF code that we have defined above is empty. Therefore any given proof is by definition very far from the code and the verifier must reject with good probability. $\square$