Lecture 6: PCPs, locally testable codes, and relation to cosystolic expansion

Irit Dinur

In this lecture we will begin with an excursion to the world of PCPs, and then describe locally testable codes (LTCs), and how this notion relates to cosystolic expansion. We will describe two notions of local testability for tensor codes: robust testability and agreement testability.

1 Probabilistically Checkable Proofs

Probabilistically Checkable Proofs are a robust form of NP proofs. Recall that the class NP is the class of all computational problems whose solutions can be efficiently checked. A solution is a "proof" that the problem is solvable, and the verifier specifies the exact rules for deciding which proofs are valid.

The study of proof systems, together with resources of randomness and interaction, has greatly developed over the past few decades, especially thanks to cryptography. One of the success stories in this area is the theory of PCPs.

A PCP is an NP proof system where the verifier is ultra efficient. The verification algorithm can be described as a random process that uses a logarithmic number of random bits and based on them reads $O(1)$ bits from the proof π. Since it doesn't read the entire proof, it cannot hope to reach the correct conclusion all of the time. Instead, given a potential $x \stackrel{?}{\in} L$, and a proof π

- (Completeness:) If $x \in L$ and π is a valid proof, the verifier always accepts.
- (Soundness:) If $x \notin L$ then for all proofs π the verifier accepts with probability at most s.

The parameter s is called the soundness error, because it is the probability of the verifier outputting the wrong answer.

Example 1.1. The language $3SAT \in \mathbf{NP}$ consists of all satisfiable 3-CNF formulae. A 3-CNF formula is specified by n variables and m clauses of the form $x \vee y \vee \neg z$ (with any negation pattern).

Given a candidate formula φ, a proof for the fact that φ is satisfiable is a truth assignment $\pi \in \{0,1\}^n$. The verifier checks that the assignment is correct by plugging in the values clause by clause.

The verifier can be made local and randomized: choose a random $i \in [m]$ and check only that the i-th clause is satisfied by the given assignment $\pi \in \{0,1\}^n$.

Completeness clearly holds: if φ is satisfiable the verifier will always except. What is the soundness error? Clearly, the natural NP verifier has soundness error that can be as large as $1 - 1/m$.

The PCP theorem [2, 1] says that every NP problem, and $3SAT$ in particular, has a verification procedure in which (a) the verifier reads $O(1)$ bits from the proof and (b) the verifier has constant soundness error.

Theorem 1.2 (PCP theorem [2, 1])**.** *There exists a constant $s > 0$ such that every $L \in$ **NP** has a polynomial-time algorithm that on input x computes a 3-CNF φ, such that,*

- *If $x \in L$ there is some $\pi \in \{0,1\}^n$ such that $\varphi(\pi) = \bar{\mathtt{T}}$*
- *If $x \notin L$ then for all $\pi \in \{0,1\}^n$, $wt(\varphi(\pi)) > 1 - s$; where we define $wt(\varphi) = \mathbb{P}_{r\in[m]}[\varphi(\pi)_r = \mathtt{F}]$.*

The algorithm mentioned in the theorem is essentially a compiler. After compilation, we get a simple verification procedure for L: the prover and verifier know the compiler algorithm. Upon input x, they both compute φ. The prover sends π as the proof, and the verifier selects a random $r \in [m]$ and checks if the rth clause of φ is satisfied under π.

There are several parameters of interest in a PCP,

- Proof length: we want the PCP proof to be efficient with respect to the original witness; as close as possible to linear size
- Query complexity: we want the verifier to read as few bits as possible from the proof
- Alphabet: the PCP proof need not be written in binary. We want small alphabet size but are willing to tolerate larger alphabets in return for fewer queries
- Soundness error: We want this to be as small as possible.

Tradeoffs between these parameters has been studied and is largely understood. A few outstanding open problems are the "sliding scale conjecture" and the "linear length PCP"; and the "unique games conjecture" which we will discuss in a future lecture.

Open Question 1.3 (sliding scale conjecture)**.** Is there a PCP verifier that has polynomially small soundness-error, perfect completeness, while making a constant number of queries to a Proof written with a polynomially Abounded alphabet size.

Open Question 1.4 (linear length PCP)**.** Is there a PCP compiler algorithm that takes as input a $3SAT$ formula φ, and computes a formula φ', whose size is linear in the size of φ, and such that:

- if $\varphi \in 3SAT$ then $\varphi' \in 3SAT$
- if $\varphi \notin 3SAT$ then every assignment falsifies a constant fraction of the clauses of φ'.

PCPs are extensively studied in two main domains. Hardness of approximation, which we discuss below; and more recently in practical cryptographic applications such as blockchains, where one needs very efficient verification.

1.1 Constraint Satisfaction Problems (CSPs)

A map $\varphi : \{0,1\}^n \to \{\mathtt{T},\mathtt{F}\}^m$ is said to be q-local if for each $i \in [m]$, the i-th output bit depends on at most q input bits. A CSP is given by a q-local map φ. It is satisfiable if $\varphi^{-1}(\bar{\mathtt{T}}) \neq \phi$. Namely, if there is an assignment that causes every constraint to evaluate to $\mathtt{T}$. Thus, every φ can be viewed as a decision problem.

Complexity theory is concerned with understanding the complexity of various problems, with relation to their syntactic "computational" structure. Constraint satisfaction problems (CSPs) provide a class of problems whose complexity is extensively studied. One successful classification of CSPs is according to their predicate.

A CSP is said to be a $\mathcal{P}$-CSP for a collection $\mathcal{P} = \{P^1, \ldots, P^T\}$ of predicates $P^t : \{0,1\}^q \to \{0,1\}$ if every output bit is the result of applying some predicate $P \in \mathcal{P}$ to some sequence of q input bits $i_1, \ldots, i_q \in [n]$.

Example 1.5. Here are a few favorite examples of CSPs in addition to $3SAT$,

- Max-CUT: given a graph G, find a partition of the vertices to two sets maximizing the number of edges that cross between the parts. One can represent each vertex by a 0/1 variable, and put a local constraint for each edge requiring the two variables to take different values.
- 3LIN: here we are given a sequence of linear equations modulo 2 over n variables, where each equation involves three variables.
- 3COL: given a graph G color the vertices using three colors, so that a maximum number of edges are bi-chromatic. One can represent each vertex by a three-valued variable, and put a local constraint for each edge requiring the two variables to take different values.

We define, for any CSP, the set of satisfying assignments,

Definition 1.6 (satisfying assignments)**.** Let $\varphi : \{0,1\}^n \to \{0,1\}^m$ be a CSP. We define

$$SAT(\varphi) = \left\{ a \in \{0,1\}^n \;\middle|\; \varphi(a) = \bar{1} \right\}.$$

This definition is equivalent to $Ker(\varphi)$ in cosystolic expansion. We only need to switch between the syntactic meaning of $'1'$ and $'0'$.

Given any CSP φ, and given a potential assignment $w \in \{0,1\}^n$ to its variables, we can look at two different measures of satisfaction:

1. The first measure, $wt(\varphi(w))$, counts how many local constraints are satisfied by the assignment.
2. The second measure, $\mathrm{dist}(w, SAT(\varphi))$, is the distance of the given assignment to the set $SAT(\varphi)$ of all satisfying assignments.

When the first measure always dominates the second measure up to a multiplicative factor h, we say that the CSP is robust:

Definition 1.7 (Robust CSPs)**.** Let $h > 0$. We say that φ is h-robust if

$$\forall w \in \{0,1\}^n, \quad \frac{wt(\varphi(w))}{\mathrm{dist}(w, SAT(\varphi))} \geqslant h.$$

Observe how similar this is to cosystolic expansion: We need to switch between $SAT(\varphi)$ and $Ker(\varphi)$.

Remark 1.8. If we have a compiler algorithm that generates, for every input x a CSP φ_x that is robust, then this implies the PCP theorem (but it is stronger).

2 Locally Testable Codes

Recall that an error correcting code is a linear subspace $C \subseteq \{0,1\}^n$. This subspace can be described by a so-called parity-check matrix $H : \{0,1\}^n \to \{0,1\}^m$ so that $C = KerH$. If H is sparse then the code is LDPC, and it is easy to compute the syndrom of a given word w.

Definition 2.1 (Locally Testable Code)**.** A code C is a locally testable code if there is an LDPC parity check matrix $X : \{0,1\}^n \to \{0,1\}^m$ that is h-robust, namely such that

$$\forall w \in \{0,1\}^n, \quad \frac{wt(Hw)}{\text{dist}(w,C)} \geqslant h.$$

For a constant h, this allows us to estimate the distance of w from the code by looking at a few bits from w.

Example 2.2 (Hadamard Code)**.** The Hadamard code consists of all functions $f : \mathbb{F}_2^n \to \mathbb{F}_2$ that are $\mathbb{F}_2$-linear, namely they satisfy

$$f(x) + f(y) = f(x+y), \qquad \forall x, y \in \mathbb{F}_2^n.$$

The above is also a good test for this code (as we have seen). Namely, if we know that 99% of these equations are satisfied, it means that f is close to a linear function.

Example 2.3 (Reed-Muller Code)**.** The Reed Muller code over $\mathbb{F}_q$ has parameters m and d and consists of all functions $f : \mathbb{F}_q^m \to \mathbb{F}_q$ that are m-variate polynomials with total degree at most d. They satisfy, for every $a, b \in \mathbb{F}_q^m$,

$$f(at+b) \text{ is a univariate polynomial (as a function of } t) \text{ of degree at most } d.$$

The above is also a good test for this code. This is the famous "low degree test". We remark that there are several variants of the low degree test depending especially on the relation between the field size q and the degree d.

Tensor codes, robust testability and agreement testability

Robust the stability is a notion of the stability that has been studied in the PCP literature. Here, we are interested not only in the fraction of local views that reject but rather in how far is the local view to an accepting one. And rather than defining this notion in full generality, let us focus on special case of tensor codes. Given two codes $C_1, C_2 \subseteq \{0,1\}^n$, the tensor code $C_1 \otimes C_2$ consists of all $n \times n$ matrices M such that

$$\begin{aligned} \forall i \in [n], \qquad & M(i,\cdot) \in C_1 \\ \forall j \in [n], \qquad & M(\cdot,j) \in C_2 \end{aligned}$$

In general, the codes need not have the same blocklength n but we focus on this case for simplicity. Here is a potential test for the question $M \stackrel{?}{\in} C_1 \otimes C_2$:

- Select with probability half a random row i, and with probability half, a random column j.
- Accept if $M(i,\cdot) \in C_1$ for a row, and if $M(\cdot,j) \in C_2$ for a column.

Robust testability is not about the probability of rejection of this test, but rather about the average distance of a local view of the tester from a valid local view. Namely, it is concerned with the average distance of a row (column)of M from C_1 (C_2). Let us define this precisely,

Definition 2.4 (Robust testability of tensor codes)**.** Fix $C_1, C_2 \subseteq \mathbb{F}_2^n$ linear error correcting codes. For $M : [n] \times [n] \to \mathbb{F}_2$, let

$$\mathrm{dist}_{col}(M) = \mathrm{dist}(M, C_1 \otimes \mathbb{F}_2^n), \qquad \mathrm{dist}_{row}(M) = \mathrm{dist}(M, \mathbb{F}_2^n \otimes C_2).$$

and

$$\begin{aligned} d(M) &= \frac{1}{2}(\mathrm{dist}_{col}(M) + \mathrm{dist}_{row}(M)). \\ &= \frac{1}{2} \mathop{\mathbb{E}}_{i \in [n]} \mathrm{dist}(M(i,\cdot), C_1) + \frac{1}{2} \mathop{\mathbb{E}}_{j \in [n]} \mathrm{dist}(M(\cdot,j), C_2). \end{aligned}$$

The robust testability of $C_1 \otimes C_2$ is defined to be

$$\rho = \min_{M \notin C_1 \otimes C_2} \frac{d(M)}{\mathrm{dist}(M, C_1 \otimes C_2)},$$

and we say that $C_1 \otimes C_2$ is ρ-robustly testable.

Robust testability is related to agreement testing, which we describe next. Suppose we are given for every column, a codeword of C_1 that is supposed to be the restriction of M to that column. The space of these matrices is denoted $C_1 \otimes \mathbb{F}_2^n$. So let this collection be called $M_{col} \in C_1 \otimes \mathbb{F}_2^n$. Similarly for every row, we are given a codeword that is supposed to be the restriction of M to that row. Denote this collection $M_{row} \in \mathbb{F}_2^n \otimes C_2$.

The probability that a random row agrees with a random column is called the agreement of M_{row} and M_{col}. It is equal to $\mathrm{dist}(M_{row}, M_{col})$. The code $C_1 \otimes C_2$ is called agreement-testable if this agreement probability upper bounds the distance of M_{row}, M_{col} from the tensor code.

Definition 2.5 (agreement testability)**.** Let $\beta > 0$. We say that $C_1 \otimes C_2$ is β-agreement testable if for every $M_{col} \in C_1 \otimes \mathbb{F}_2^n$, $M_{row} \in \mathbb{F}_2^n \otimes C_2$, there exists $w \in C_1 \otimes C_2$ such that

$$\beta \cdot (\mathop{\mathbb{P}}_{i}[M_{row}(i,\cdot) \neq w(i,\cdot)] + \mathop{\mathbb{P}}_{j}[M_{col}(\cdot,j) \neq w(\cdot,j)]) \leqslant \mathop{\mathbb{P}}_{i,j}[M_{col}(i,j) \neq M_{row}(i,j)].$$

Lemma 2.6. *If $C_1 \otimes C_2$ is β-agreement testable, then $C_1 \otimes C_2$ is τ-robustly testable for $\tau = \frac{\beta}{2(\beta+1)}$.*

Moreover, if $C_1 \otimes C_2$ is τ-robustly testable then $C_1 \otimes C_2$ is β-agreement testable, for $\beta = \frac{2\tau\delta_1\delta_2}{\delta_2 + \delta_1(1+2\tau)}$ (where δ_i is the relative distance of C_i).

Proof. Given M, define M_{row}, M_{col}. By assumption they are close to M, so to each other, so have high agreement. □

We will see at a later point in the course how this relates to coboundary expansion of the chain

$$\mathbb{F}_2^{k^2} \xrightarrow{\delta_0} \mathbb{F}_2^{2nk} \xrightarrow{\delta_1} \mathbb{F}_2^{n^2} \tag{2.1}$$

where $\delta_0 = G_1 \otimes I + I \otimes G_2$ and $\delta_1 = I \otimes G_2 + G_1 \otimes I$ for G_1, G_2 the generating matrix of C_1, C_2.

The property of robustness of a tensor code depends on the component codes, as can be seen in the following example.

Example: tensor of two Reed-Solomon codes

Let $C_1 = C_2$ be the Reed-Solomon code with field size $n = p$ and degree d. Polyschuk and Spielman [3] showed that if $d < p(\frac{1}{2} - \varepsilon)$ then the tensor of two RS_d codes is ε^2 robust. What happens when d gets closer to $p/2$? It turns out that the tensor code is no longer robust, as can be seen in the following example due to S. Kopparty. Let $f(x, y) = 1 - (x - y)^{p-1}$. Then over $\mathbb{F}_p$ this function equals 1 if $x = y$ and zero otherwise because $a^{p-1} = 1$ for all $a \neq 0$ in $\mathbb{F}_p$. On the other hand, by setting $j = p - 1 - i$, it can be written as

$$f(x, y) = \sum_{i<p/2} x^i y^{p-1-i} + \sum_{j<p/2} y^j x^{p-1-j} = M_{row}(x, y) + M_{col}(x, y)$$

where $M_{row}(\cdot, y)$ has degree below $p/2$ for each $y \in \mathbb{F}_p$, and $M_{col}(x, \cdot)$ has degree below $p/2$ for each $x \in \mathbb{F}_p$. This gives a concrete counter example for the testability of the tensor of two Reed-Solomon codes when $d = p/2$. Indeed, M_{col}, M_{row} have very high agreement probability: $1 - 1/n$, and yet their distance to the tensor code is $\Omega(if1)$.

References

[1] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and intractability of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. 2

[2] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. 2

[3] Alexander Polishchuk and Dan Spielman. Nearly linear size holographic proofs. In *Proc. 26th ACM Symp. on Theory of Computing*, pages 194–203, 1994. 6