Lecture 3

Lecturer: Irit Dinur
Preliminary Scriber: Karthik C. S.
Scribe Upgrader: Inbal Livni

We revisit the connections between PCPs and High dimensional expanders. In the coming lectures we will see that the previously defined Ramanujan expanders are indeed cosystolic expanders. The direction that we hope to prove to be true is that these high dimensional complexes can be useful for constructing better PCPs and LTCs. More concretely, we hope to achieve the following:

- Short(er) PCPs. We would like to have shorter PCP proofs for languages in **NP**.
- PCPs with a large gap. This would give tight inapproximability results, and might help in better addressing the unique games conjecture, etc.
- More explicit PCPs. The construction of Ramanujan graphs is explicit, whereas the current PCP constructions are recursive.

We would also like to relate agreement expansion with coboundary expansion (or even cosystolic expansion). Recollect that the PCP theorem gives us a polynomial time algorithm taking an input graph G and outputting a graph H such that if $\mathrm{val}(G) = 1$ then $\mathrm{val}(H) = 1$ and if $\mathrm{val}(G) < 1$ then $\mathrm{val}(H) < 0.999$. In the last lecture we saw that starting from G, we had an intermediate transformation to an agreement problem on a complex X. We remark here that we would perform an "alphabet reduction" to reach H from X, but discussing more about this step is out of the scope of this workshop.

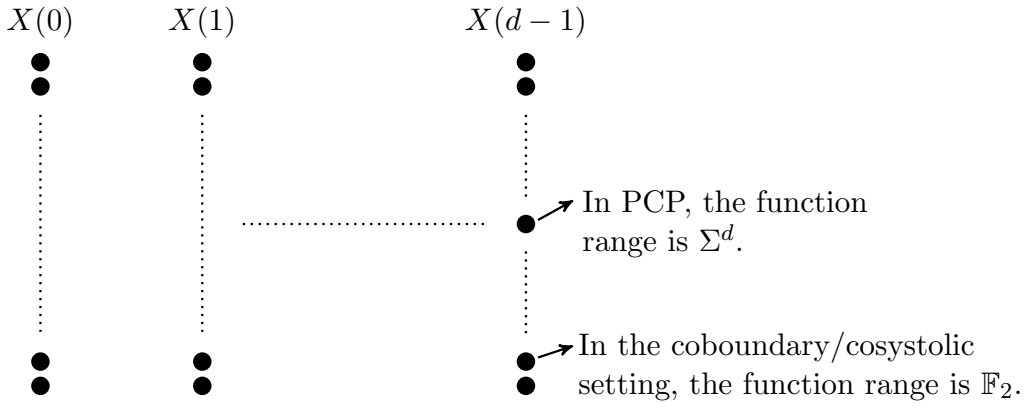

Figure 1: X is a simplicial complex.

We note that if we had an explicit coboundary expander then we would not even need the intermediate step of "alphabet reduction" because each assignment to a simplex can be easily seen as a concatenation of symbols over the smaller alphabet, while *maintaining* the consistency

of the constraints imposed on the simplex.

We would now like to address how to obtain a large gap (1 vs 0.01) from a small gap (1 vs 0.99). We care about getting large gaps because, then we could use the known gadget replacement/reductions from literature to obtain tight hardness of approximation bounds. Elaborating, let X be a complex and $f \in \mathfrak{a}_{k-1}(X, \Sigma)$. Suppose we have $\alpha \geq 0.01$ then what can we say about f? We will see in the following lecture that if $\alpha \geq 0.99$ then we **can** say something about the structure of f. We can think of these problems/questions as "inverse problems" for agreement or as understanding "strong expansion".

We will move to a new topic, and discuss about hardness amplification. Given a Boolean function $\varphi : \{0,1\}^n \to \{0,1\}$, it is called δ-hard for a class of functions $\mathcal{C}$ if the following holds:

$$\max_{f \in \mathcal{C}} \Pr_{x \in \{0,1\}^n} [f(x) = \varphi(x)] \leq 1 - \delta.$$

The goal is to construct $\varphi' : \{0,1\}^{n'} \to \{0,1\}$ that is $\frac{1}{2} - \varepsilon$-hard for some class $\mathcal{C}'$. Note that one of the constant functions **0,1** is $\frac{1}{2}$ close to any Boolean function.

Consider $\varphi^{\otimes k} : \{0,1\}^{nk} \to \{0,1\}^k$ defined by $\varphi(\bar{x}_1, \ldots, \bar{x}_k) = (\varphi(\bar{x}_1), \ldots, \varphi(\bar{x}_k))$. It seems like $\varphi^{\otimes k}$ is $1 - (1-\delta)^k$-hard, although this function isn't Boolean. We call $\varphi^{\otimes k}$ as the direct product.

Consider $\varphi' \{0,1\}^{nk+k} \to \{0,1\}$ defined as follows:

$$\begin{aligned} \varphi(\bar{x}_1, \ldots, \bar{x}_k, r) &= \sum_i r_i \varphi(\bar{x}_i) \mod 2 \\ &= \langle \varphi^{\otimes k}(x_1, \ldots, x_k), r \rangle \mod 2, \end{aligned}$$

it seems like φ' is $\frac{1}{2} - \varepsilon$-hard for $\varepsilon = (1-\delta)^k$. The general idea in hardness amplification is that if $\varphi^{\otimes k}$ or φ' can be computed by $\mathcal{C}$ on more than ε (or $\left(\frac{1}{2} + \varepsilon\right)$) fraction of the inputs then we will construct a $\mathcal{C}'$ that computes φ on more than $1 - \delta$ fraction of the inputs. In hardness amplification of $\varphi^{\otimes k}$ we look at $X(0)$, the vertices of the simplicial complex, as the domain of φ and at $X(k-1)$ as the domain of $\varphi^{\otimes k}$, we will discuss more about it below.

We will now see local approximate list decoding which is the work of [IJKW10]. While they do not talk of their constructions in the language seen in this workshop, they indeed build structures such as the intersection code, which are closely related to the agreement problem on complexes that we have seen. Elaborating, for any complex X, and alphabet $\Sigma = \{0,1\}$, we first recollect the following definitions

$$\mathfrak{a}_0(X) = \{f : X(0) \to \{0,1\}\},$$

$$\mathfrak{a}_i(X) = \{f : X(i) \to \{0,1\}^{i+1}\},$$

and $U^i : \mathfrak{a}_0 \to \mathfrak{a}_i$, such that for each $f \in \mathfrak{a}_0$, $U^i f(S) = (f(x))_{x \in S}$.

Then we define the code of a complex X:

$$\text{code}(X) = U^{k-1}\mathfrak{a}_0 = \{U^{k-1} f \mid f : X(0) \to \{0,1\}\}.$$

Note that $U^{k-1}\mathfrak{a}_0 \neq \mathfrak{a}_{k-1}$. The goal will be related to the decoding the above code. We would like to prove the following result when X is the complete complex.

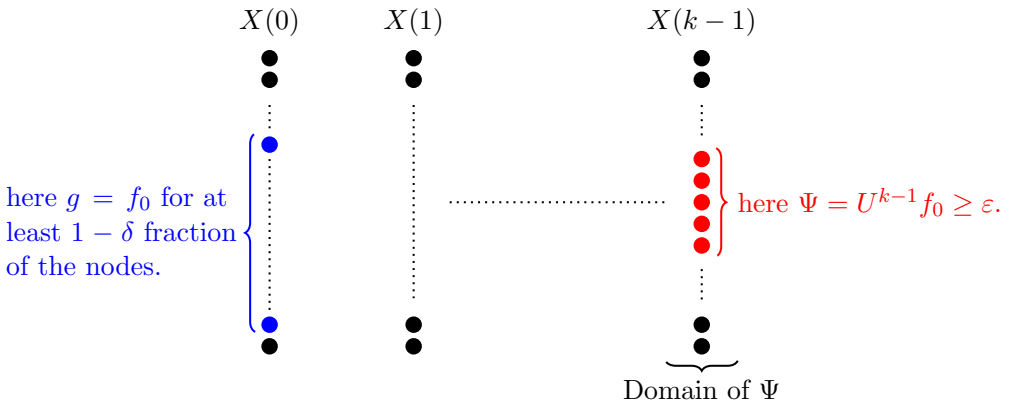

Figure 2: ψ is ε close to a code, g is a function on $X(0)$ that is $1-\delta$ close to f_0.

Theorem 1. *Let $\psi \in \mathfrak{A}_{k-1}$ be ε-close to code(X), i.e., there is some $f_0 \in \mathfrak{A}_0$ such that $\Pr_{S \in X(k-1)}[\psi(S) = U^{k-1} f_0(S)] \geq \varepsilon$. Moreover, let ψ be computed by a small circuit. Then, there is an efficient algorithm* computing g such that $\Pr_{x \in X(0)}[f_0(x) = g(x)] \geq 1-\delta$.*

This theorem amplifies the hardness: if it is hard to compute f_0 on $1-\delta$ fraction of the inputs, then it is hard to compute ψ even on ε fraction of the tuples.

We remark here that even though small circuits can compute majority, just computing majority is not enough in this case, since on most sets $\psi(S)$ can give a bad value. Moreover, there can be $\frac{1}{\varepsilon}$ different functions f_0, all satisfying the theorem conditions for the same ψ.

Proof of Theorem 1. We will output a list of algorithms (i.e., circuits) $C_1, \ldots C_L : X(0) \to \{0,1\}$ such that for every $f_0 \in \mathfrak{A}_0$, $U^{k-1} f_0$ ε-agrees with ψ, and there is some $i \in [L]$ such that $\Pr_x[C_i(x) = f_0(x)] \geq 1-\delta$.

The idea is to "guess and check". More formally,

1. Choose $T \in X(t-1)$ at random.
2. Guess $f_0(u)$ for all $u \in T$.
3. For every $\sigma \in \{0,1\}^T$, C_σ is our list that will outputted in the end.

Let $L(T) = \{S \in X(k-1) \mid T \subseteq S\}$ and $L_\sigma(T) = \{S \in L(T) \mid \psi(S)_T = \sigma\}$. We have the following definition of "good" T, assuming we already have the correct σ:

Definition 2. *T is good if $\Pr_S[S \in L_\sigma(T) \mid S \in L(T)] \geq \frac{\varepsilon}{2}$.*

We have the following algorithm on input x.

1. If $x \in T$ then output $\sigma(x)$.
2. Else, choose at random some S which contains $\{x\} \cup T$.
   (a) If $\psi(S)_T = \sigma$ then output $\psi(S)(x)$.
   (b) Else, repeat this procedure of picking S.

We note the following upper bound on the number of iterations of the above algorithm:

*has to be a probabilistic algorithm

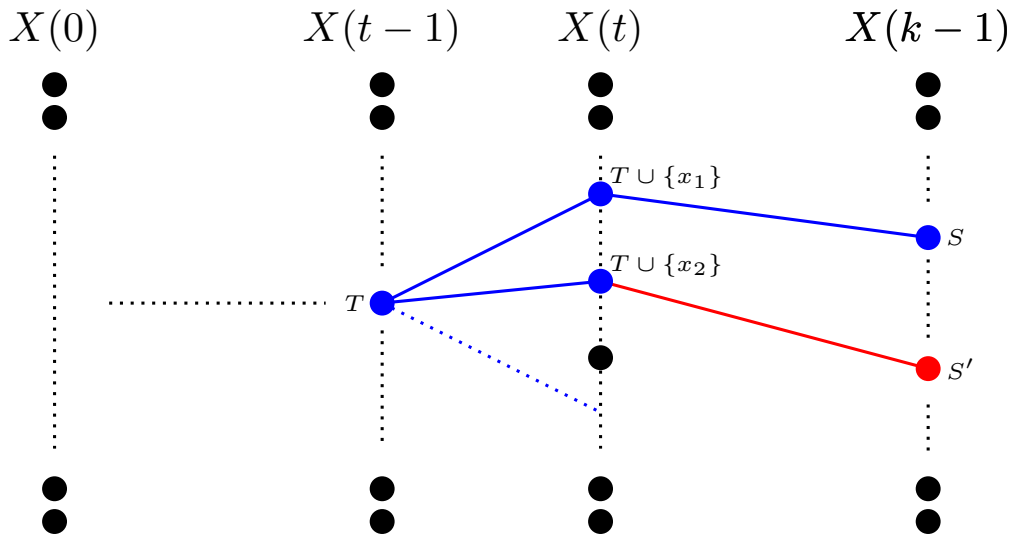

Figure 3: An edge is bad if $\psi(S')_T = \sigma$, but $\psi(S')_{x_2} \neq f_0(x_2)$, the bad edge is red and the rest are blue.

Claim 3. *If T is good then for almost all $x \notin T$ also, $T \cup \{x\}$ "is good", i.e.,*

$$\Pr_{S \in X(k-1)}[\psi(S) = U^{k-1} f_0(S) \mid x, T \in S] \geq \frac{\varepsilon}{3}.$$

Consequently, the algorithm will halt after poly $\left(\frac{1}{\varepsilon}\right)$ steps. We remark that the above claim can also been seen as the sampling properties of the complete complex.

For each S, let $\mathrm{err}(S) = \Pr_x[f_0(x) \neq \psi(S)(x) \mid x \in S]$. We make a better definition to look at "excellent" T:

Definition 4. *T is excellent if $\mathbb{E}_{S \in L_\sigma(T)}[err(S)] < \frac{1}{100}$.*

The main remaining claim then would be to say that a good T is usually also excellent. □

References

[IJKW10] Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi Wigderson. Uniform direct product theorems: simplified, optimized, and derandomized. *SIAM Journal on Computing*, 39(4):1637–1665, 2010.