CMPUT 675: Computational Complexity Theory **Winter 2019**

Lecture 20 (March 21): Construction of Expanders and PCP Main Lemma

Lecturer: Zachary Friggstad *Scribe: Ramin Mousavi*

20.1 Finishing the Proof of the Expander Walk Theorem

We give a quick overview of the definitions from the previous lecture, see the last lecture for more precise definitions.

Recall the definition of a random walk matrix A_G of a d-regular graph $G = (V, E)$. Let $|\lambda_1(A_G)| \geq |\lambda_2(A_G)| \geq ... \geq |\lambda_n(A_G)|$ be the eigenvalues of A (with multiplicity), we omit the subscript G when the graph is clear from the context. We say G is an (n, d, λ)-expander if $|V| = n$, G is a d-regular graph, and $|\lambda_2(A)| \leq \lambda$ (in fact, we can show that $\lambda_2(A)$ is bounded at least zero so we do not need the absolute value). Recall from the last lecture that the Kronecker product of two matrices A and B is denoted by $A \otimes B$. The spectral norm of a matrix A is denoted by $\|A\|$ and is the maximum stretch over the unit vectors. Denote the n-dimensional vector with $\frac{1}{n}$ on all its coordinates by $\mathbf{1}$.

In the proof of expander walk theorem from last lecture, we used Lemma 1 (below) without proving it. We prove this lemma in this section. In Section 20.2, we discuss some tools that will be used in Section 20.3 for constructing a family of expander graphs. In Section 20.4, we state the main lemma for proving the PCP theorem and show how the PCP theorem follows from this lemma.

Lemma 1 *Let G be an (n, d, λ)-expander graph, and let A be the random walk matrix of G. Then*

$$A = (1 - \lambda)J + \lambda C, \tag{20.1}$$

where J_n is the random walk matrix of a complete graph with loops and C is a matrix with $\|C\| \leq 1$

Proof. We assume $\lambda \neq 0$, one can prove that if $\lambda = 0$ (i.e. all eigenvalues are 0 except for the first eigenvalue) then the random walk matrix has $1/n$ in each entry: indeed any vector v that is orthogonal to 1 would then be an eigenvector so it would have $A \cdot v = \mathbf{0}$. Since this holds for all $v \perp \mathbf{1}$, then every row of A has all entries being the same. As each row of of A sums to 1, then A has $1/n$ in each entry.

Define C to be $C := \frac{1}{\lambda}(A - (1 - \lambda)J_n)$. We claim that $\|C\| \leq 1$. Let x be a unit vector with $x = u + v$, where $u = \alpha \cdot \mathbf{1}$ for some α and $v \perp \mathbf{1}$. Note that $\mathbf{1}$ is the eigenvector corresponding to the largest eigenvalue (which is 1) of both A and J_n, and hence $Cu = u$. Also $J_n v = 0$ and so $Cv = \frac{1}{\lambda}Av$. By a Rayleigh quotient, together with the facts that $v \perp \mathbf{1}$ and $\lambda_2 \leq \lambda$, we have

$$\|Av\|_2^2 \leq \lambda_2^2 \|v\|_2^2 \leq \lambda^2 \|v\|_2^2. \tag{20.2}$$

Using the above observations, we get

$$\|Cx\|_2^2 = \|Cu + Cv\|_2^2 = \|u + \frac{1}{\lambda}Av\|_2^2 = \|u\|_2^2 + \frac{1}{\lambda}\|Av\|_2^2 \leq \|u\|_2^2 + \|v\|_2^2 = \|u + v\|_2^2 = \|x\|_2^2 = 1,$$

where the third equality holds by Pythagoras Theorem and the fact that $Av \perp \mathbf{1}$[1], the first inequality comes from (1.2), the second to the last equality holds by Pythagoras Theorem and the fact that $u \perp v$. ■

[1] This follows if we write v as a linear combination of eigenvectors of A.

20.2 Different Graph Products

In this section, we introduce two graph products and explore their properties. We start with an easy one.

20.2.1 Path Product

Let G be an (n, d, λ)-expander with random walk matrix A. For $t \geq 2$, the *path product* of G (t times) is denoted by G^t is the graph described by the random walk matrix A^t. It is easy to see that $A^t_{i,j}$ is the number of walks of length exactly t in G from i to j divided by d^t. Since, for each vertex v, there are exactly d^t many different walks of length t starting at v, G^t is d^t-regular. Furthermore, using the definition of eigenvalue, we can see that $\lambda_i(A^t) = (\lambda_i(A))^t$, and hence $|\lambda_2(A^t)| \leq \lambda^t$. We conclude that G^t is (n, d^t, λ^t)-expander.

20.2.2 Replacement Product

Let G be a D-regular graph on n vertices with random walk matrix A, and let H be a d-regular graph on D vertices with random walk matrix B. The *replacement product* of G and H is denoted by $G \circledR H$ is a $2d$-regular graph with $n \cdot D$ vertices constructed as follows:

Label vertices of H by integers in $\{1, ..., D\}$. For each vertex u of G, we label the neighbours of u by a number in $\{1, ..., D\}$, please see Example 1.

1. Replace each vertex $v \in V(G)$ by a copy H_v of H.
2. For each $uv \in E(G)$, let v be the i-th neighbour of u, and let u be the j-th neighbour of v (based on our labeling defined above). Then, place d parallel edges between vertex i in H_u and vertex j in H_v.

Let us compute the random walk matrix M of $G \circledR H$ in terms of A and B. Define $\hat{A}$ to be an $(n \cdot D) \times (n \cdot D)$ matrix where its rows and columns are indexed by pairs (u, i) where $u \in V(G)$ and $i \in V(H)$. The (v, j)-th column of $\hat{A}$ has 0 everywhere except a single 1 on the (u, i)-th place where v is the i-th neighbour of u and u is the j-th neighbour of v. So $\hat{A}$ represent the edges corresponding to the edges in G (i.e., the edges that were obtained by step 2).

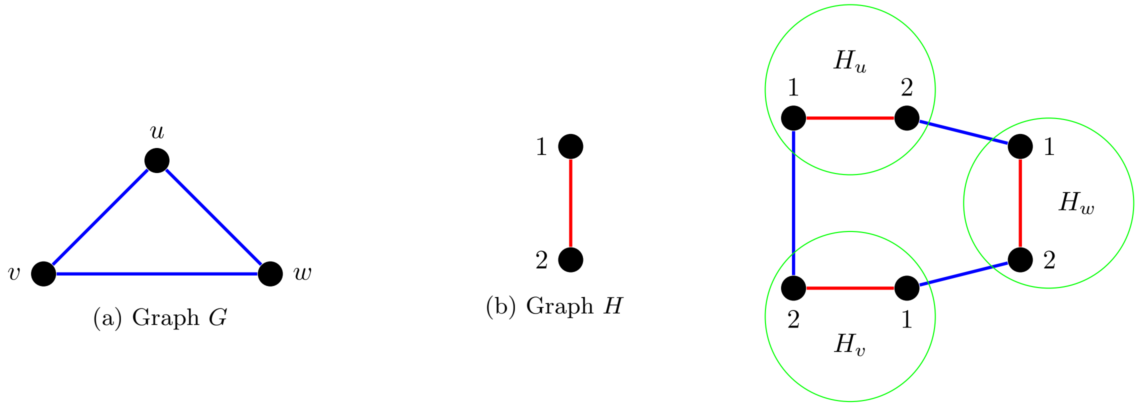
It is easy to see that $I_n \otimes B$ represent the edges corresponding to the edges in H (i.e., the edges that were obtained by step 1).

The use of parallel edges in step 2 is to make sure that from a vertex in $G \circledR H$, with the same probability either we walk on an edge corresponding to $E(H)$ or on an edge corresponding to $E(G)$. Thus, we can describe M as follows:

$$M = \frac{1}{2}\hat{A} + \frac{1}{2}(I_n \otimes B). \tag{20.3}$$

When $d << D$, the replacement product results in a graph with substantially smaller degree than G's degree while its expansion is not much less than G's expansion. The former statement is clear, but we formalize the latter statement in Lemma 2. Before stating the lemma, let us give an example of replacement product. Note that the following example is merely for clarifying the construction of this product and we did not take in to account the expansions.

Example 1 *Let G and H be the two graphs defined in Figure 20.1. Consider the following labeling of neighbours of vertices in G:*



(a) Graph G

(b) Graph H

(c) Graph $G \circledR H$

Figure 20.1: Graphs for Example 1.

- *Labeling of the neighbours of u: v has label 1, and w has label 2.*
- *Labeling of the neighbours of v: w has label 1, and u has label 2.*
- *Labeling of the neighbours of w: u has label 1, and v has label 2.*

Then, $G \circledR H$ is the graph shown in Figure 20.1(c), respect to the above labeling.

Lemma 2 *Let G be an $(n, D, 1-\epsilon)$-expander graph, and let H be an $(D, d, 1-\delta)$-expander graph. Then, $G \circledR H$ is an $(n \cdot D, 2d, 1 - \frac{\epsilon \cdot \delta^2}{24})$-expander graph.*

Proof. Let A, B, and M be the random walk matrices for G, H, and $G \circledR H$, respectively. By Lemma 1, we can write

$$B = \delta J_D + (1-\delta)B', \tag{20.4}$$

where $\|B'\| \leq 1$. We need to show that $\lambda_2(M) \leq 1 - \frac{\epsilon\delta^2}{24}$ which suffices to show that $\lambda_2(M^3) = (\lambda_2(M))^3 \leq 1 - \frac{\epsilon\delta^2}{8}$. Using the LHS of (20.4) instead of B in M, we get

$$M^3 = (\frac{1}{2}\hat{A} + \frac{\delta}{2}(I_n \otimes J_D) + \frac{1-\delta}{2}(I_n \otimes B'))^3 \tag{20.5}$$

$$= (1 - \frac{\delta^2}{8})C + \frac{\delta^2}{8}(I_n \otimes J_D)\hat{A}(I_n \otimes J_D), \tag{20.6}$$

where $\|C\| \leq 1$. This is because, all the matrices involved in (1.5) have spectral norm at most 1, and the fact that both $\|AB\|$ and $||A \otimes B||$ are bounded by $\|A\|\|B\|$ for any two symmetric matrices A, B (this is left as an exercise). It is clear that $\|I_n\| = \|J_D\| = 1$. Also $\|\hat{A}\| = 1$ because of its definition $\hat{A}$ is a permutation matrix of I_{nD}. Finally, since $\|B'\| \leq 1$ (by Lemma 1), we have $\|I_n \otimes B'\| \leq 1$ (this is left as an exercise).

We will prove the following:

Claim 1 $(I_n \otimes J_D)\hat{A}(I_n \otimes J_D) = A \otimes J_D$.

Suppose the claim holds for a moment, together with (20.6), for any x such that $x \perp \mathbf{1}$, we have

$$\begin{aligned}
\|M^3x\|_2 &\leq (1-\frac{\delta^2}{8})\|Cx\|_2 + \frac{\delta^2}{8}\|(A\otimes J_D)x\|_2 && (20.7)\\
&\leq (1-\frac{\delta^2}{8})\|x\|_2 + \frac{\delta^2}{8}\max\{\lambda_2(A), \lambda_2(J_D)\}\|x\|_2 && (20.8)\\
&= (1-\frac{\delta^2}{8})\|x\|_2 + \frac{\delta^2}{8}\lambda_2(A)\|x\|_2 && (20.9)\\
&= (1-\frac{\delta^2}{8})\|x\|_2 + \frac{\delta^2}{8}(1-\epsilon)\|x\|_2 && (20.10)\\
&= (1-\frac{\epsilon\delta^2}{8})\|x\|_2, && (20.11)
\end{aligned}$$

where (20.8) follows from the facts that $\|C\| \leq 1$, and by Rayleigh quotient we have $\|(A\otimes J_D)x\|_2 \leq \lambda_2(A\otimes J_D)\|x\|_2$ for all $x \perp 1$. Again, using Rayleigh quotient, together with (20.11), we conclude that $\lambda_2(M^3) \leq 1-\frac{\epsilon\delta^2}{8}$, as desired.

It remains the proof of Claim 1.

Proof of Claim 1. We can view the matrix on the LHS as the random walk matrix of the graph on nD vertices such that from a vertex (u, i), first uniformly at random picks a label $1 \leq k \leq D$. Let v be the k-th neighbour of u in G. Then uniformly at random it picks another label $1 \leq j \leq D$ and then move to the vertex (v, j).
On the other hand, the matrix on the RHS is the random walk matrix of a graph on nD vertices such that from a vertex (u, i), first with probability $\frac{1}{D}$ picks a neighbour v of u in G and then uniformly at random picks a label $1 \leq j \leq D$ and move to (v, j).
As we showed, both matrices are describing the same random walk matrices of a graph.

■

■

20.3 Constructing Expander Graphs

We need the following lemma (Theorem 21.8 in [AB09]) about existence of expander graphs with "small" number of vertices. We omit its proof here.

Lemma 3 *For some constant $d \geq 3$, there is a $(D, d, 0.01)$-expander graph where $D = (2d)^{50}$.*

For the curious, a proof can be found in [HLW06], in particular Theorem 7.5 would imply the existence of such a graph. Then, if one wants to complete the entire description of how to construct expanders, they can just hard code this graph into the algorithm (or one could find it in $O(1)$ time using brute force, given that we know it exists).

For an appropriate constant d, the above lemma states there is an expander graph with a constant number of vertices. Thus, we can find such graph H with brute force search. Set $G_1 := H$ with edges doubled. We construct the following family of expanders:

$$G_k = G_{k-1}^{50} \circledR H. \tag{20.12}$$

Claim 2 *G_k is $(D^k, 2d, 0.98)$-expander graph, where d and D are from Lemma 3.*

Proof. We proceed by induction on k. So suppose G_{k-1} is $(D^k, 2d, 0.98)$-expander. From the property of path product that were discussed in Section 20.2.1, we know that G^{50}_{k-1} is $(D^{k-1}, (2d)^{50}, 0.98^{50})$-expander. Note that $D = (2d)^{50}$; hence, G^{50}_{k-1} Ⓡ H is defined. Recall that H is $(D, d, 0.01)$-expander. By Lemma 2, we conclude that G_k has $D^{k-1} \cdot D = D^k$ vertices, its degree is $2d$, and its expansion parameter λ is at most

$$1 - \frac{(1-0.98^{50})(1-0.01)^2}{24} \leq 0.98, \tag{20.13}$$

as desired. ■

Note that we can compute the adjacency matrix of G_k in poly(D^k). Suppose computing the adjacency matrix of G_k takes $T(k)$ time. Then, $T(k) = T(k-1) + (D^{k-1})^c$ for some constant c (the second term comes from powering a $D^{k-1} \times D^{k-1}$ matrix to a constant number and at the end computing the replacement product which can be done in poly(D^{k-1}), see (20.3)). Hence, our family of expanders is explicit.

We can get a strongly explicit expander family by introducing one further (simple) operation, see Section 21.3.5 of [AB09] for details if you are curious.

20.4 PCP Main Lemma

We now turn to discussing how to to prove the PCP theorem. Much more discussion is found in the notes of the next lecture, but we quickly summarize what was covered in this lecture.

In this section, we state the main lemma for proving the PCP theorem. Before that, we need the following definitions:

Definition 1 ($\mathbf{q}\,$CSP$_\mathbf{W}$) *For integers q, $W \geq 1$, the $q\,\mathrm{CSP}_W$ is a language consists of variables $x_1, ..., x_n$ that take value in $\{0, ..., W-1\}$ and m functions $f_j : \{0, ..., W-1\}^q \to \{0,1\}$ for $1 \leq j \leq m$.*
Let $\bar{x}$ be an assignment to the variables of an instance ϕ of $q\,\mathrm{CSP}_W$. Then, define $\mathrm{sat}(\phi)$ to be

$$\mathrm{sat}(\phi) := \max_{\bar{x}} \frac{\sum_j f_j(\bar{x})}{m}.$$

Definition 2 (CL-reduction) *A function $f : q\,\mathrm{CSP}_W \to q\,\mathrm{CSP}_W$ is a* CL-reduction *if*

- *f is polytime computable function.*
- *For a $q\,\mathrm{CSP}_W$ instance ϕ, if $\mathrm{sat}(\phi) = 1$, then $\mathrm{sat}(f(\phi)) = 1$.*
- *The number of constraints (functions) in the mapped instance increases by an $O(1)$-factor.*

Recall the PCP theorem from Lecture 14, i.e., $\mathbf{PCP}(O(\log n), O(1)) = \mathbf{NP}$. We show that the PCP theorem follows from the following lemma:

Lemma 4 (PCP main lemma) *There exist constants $q \geq 3$, $\epsilon_0 > 0$, and a CL-reduction $f : q\,\mathrm{CSP}_2 \to q\,\mathrm{CSP}_2$ such that*

$$\mathrm{sat}(\phi) = 1 - \epsilon \Rightarrow \mathrm{sat}(f(\phi)) \leq 1 - \min\{\epsilon_0, 2\epsilon\}.$$

Now the PCP theorem follows by: Let $L \in \mathbf{NP}$, then there is a Karp-reduction g from L to $q_0\,\mathrm{SAT}$, a special case of $q_0 - \mathrm{CSP}_2$. For an instance x of L, say $g(x)$ has m constraints. Apply f from the PCP main lemma for $k := \lceil \log_2 m \rceil$ times to g, i.e., compute $f^k(g(x))$. Note that from the third property of CL-reduction and the value of k, we conclude that $f^k(g(x))$ runs in polytime. Furthermore, if $x \in L$, then $f^k(g(x))$ is satisfiable. Suppose $x \notin L$. Then, $\mathrm{sat}(g(x)) \leq 1 - \frac{1}{m}$. Thus, $\mathrm{sat}(f^k(g(x))) \leq 1 - \min\{\epsilon_0, m \cdot \frac{1}{m}\} = 1 - \epsilon_0$.

From this, we get a simple $\mathbf{PCP}(O(\log n),\ O(1))$-verifier for L, i.e., given x, the proof consists of $f^k(g(x))$ and a truth assignment for $f^k(g(x))$. Then the verifier samples a random clause and check the given assignment satisfies this clause or not. In the No-case, the probability that the clause is satisfied is at most $1 - \epsilon_0$. Since $1 - \epsilon_0$ is a constant, we can reduce it to $\frac{1}{2}$ by a constant number of repetition.

References

AB09 S. Arora and B. Barak, Computational Complexity: A Modern Approach, *Cambridge University Press*, 2009.

HLW06 S. Hoory, N. Linial, and A. Wigderson, Expander graphs and their applications, Bulletins of the American Mathematical Society, 43:439–561, 2006.
`http://www.cs.huji.ac.il/~nati/PAPERS/expander_survey.pdf`