Lecture 15: Random walks and expander graphs ⋆

Rajat Mittal

IIT Kanpur

Random walks have been used in multiple areas of computer science to give randomized algorithms. In this lecture note, we will analyze the properties of random walk using the eigenvalues and eigenvectors of the Laplacian matrix. It will be shown that by slightly tweaking the natural random walk on a graph we can make sure that the probability distribution converges to the first eigenvector.

Later in the lecture, we will analyze random walk on expander graphs. Using this analysis, we will show that we can reduce the amount of randomness needed to decrease the error probability of a randomized algorithm. Lastly, we will study some properties of expander graphs.

Before we define random walk, let us remind ourselves of the notion of a randomized algorithm.

A randomized algorithm allows the creator of the algorithm to use randomness (fair coin-tosses) at various steps of the algorithm. Though, *for every input*, the answer should be correct with high probability over the randomness of the algorithm.

High probability depends on the context, but if not mentioned we will take it to be more than $\frac{2}{3}$.

1 Random walks

Suppose we are given a graph with two nodes s and t. We would like to know if these two nodes are connected or not.

A very natural approach would be to start with s and move to a neighbour of s in the next time step randomly. We can continue exploring the graph this way for some time and hope that we will reach t soon.

Exercise 1. What should the number of time steps depend on?

A random walk on a set of nodes V (say $|V| = n$) is defined to be the process where we move from node to node with pre-defined probabilities (starting state can be chosen arbitrarily). The probabilities can be captured by an $n \times n$ matrix W, where i-th row gives the transition probabilities for moving from i to other nodes.

The state of a random walk at time t is the collection of probabilities of being at various nodes i. In other words, a valid state is a non-negative vector $w \in \mathbb{R}^n$, s.t., the entries of w sum up to 1. We will call such vectors *state vectors*.

We will be interested in random walks on a graph, where V will be the vertex set. At every step, we will move from a vertex to its neighbours uniformly.

Exercise 2. Show that the walk matrix W for a d-regular graph is $\frac{1}{d}A_G$.

We will look at the case of d-regular unweighted simple graphs for simplicity. Though, notice that random walks on graphs can have weights, self loops and multiple edges. All these can be captured by changing the walk matrix W.

Let us call the starting state of the random walk to be w_0. Let us say that after t time steps, the state is w_t.

Exercise 3. Show that $w_t = W^t w_0$.

We are interested in the properties of w_t. For our case, we know that $W = \frac{1}{d}A_G$, implies that all eigenvalues of W are between 1 and -1.

⋆ The content of these notes is largely taken from Dan Spielman's course notes

1.1 Applying an operator repeatedly

From the discussion in the previous section, we are interested in applying an operator (W in our case) multiple times and study its properties. For this section, let us take a general symmetric $n \times n$ matrix M with eigenvalues between -1 and 1. Let us also assume that the eigenspace corresponding to eigenvalue 1 is one dimensional.

Notice that a symmetric matrix can always be scaled so that its eigenvalues will lie between -1 and 1.

Exercise 4. Show that W satisfies the properties mentioned above.

Suppose M has eigenvalues $\mu_1 = 1 \geq \mu_2 \cdots \geq \mu_n$ with eigenvector w_i for eigenvalue μ_i. We will denote the first eigenvector w_1 as s (corresponding to eigenvalue 1).

We will start with an arbitrary vector $u \in \mathbb{R}^n$. The task is to study $M^t u$ for a large natural number t. First, we can write u in the orthonormal basis of w_i's.

$$u = \sum_i \alpha_i w_i.$$

Since w_i's are eigenvectors and M is a linear operator,

$$M^t u = \sum_i \alpha_i M^t w_i = \sum_i \alpha_i \mu_i^t w_i.$$

Separating out the first eigenvector,

$$M^t u = \alpha_1 s + \sum_{2 \leq i \leq n} \alpha_i \mu_i^t w_i.$$

For large t, this quantity tends to $\alpha_1 s$ if $|\mu_i| < 1$ for all i not equal to 1. This will happen in almost all cases except when $\mu_n = -1$.

In other words, $M^t u$ tends to $\alpha_1 s$ if μ_n is greater than -1. We also want to know about the rate of this convergence.

We look at the quantity,

$$M^t u - \alpha_1 s = \sum_{2 \leq i \leq n} \alpha_i \mu_i^t w_i.$$

It is clear that the quantity on the right hand side goes to zero. The rate should depend on the absolute values of μ_i's. Since there is no constraint on u, rate of convergence will depend upon the μ_i with highest absolute value. Assume that μ_2 has the highest absolute value (more than μ_n).

Note 1. we have assumed that $|\mu_2| \geq |\mu_n|$, the other case can be handled similarly with μ_2 replaced by $|\mu_n|$.

Then,

$$\left\| M^t u - \alpha_1 s \right\|^2 = \sum_{2 \leq i \leq n} \left\| \alpha_i \mu_i^t w_i \right\|^2 \leq \mu_2^t \sum_{2 \leq i \leq n} \left\| \alpha_i w_i \right\|^2.$$

Exercise 5. Why is the first equality true?

Since $\|u\|^2 = \sum_i \|\alpha_i w_i\|^2$, we can simplify our expression,

$$\left\| M^t u - \alpha_1 s \right\|^2 \leq \mu_2^t \|u\|^2.$$

We know that μ_2^t will converge to zero and the rate depends upon,

$$\mu_2^t = (1 - (1 - \mu_2))^t \approx e^{-t(1-\mu_2)}.$$

So, this quantity will be $O(1/n)$ after $O(\log(n)/(1 - \mu_2))$ applications of M.

1.2 Stationary distribution and convergence

Stationary distribution for an operator M is defined to be a state vector u, such that, $Mu = u$. Remember that a state vector u is entry-wise non-negative and $\sum_i u_i = 1$. Notice that u needs to be an eigenvector with eigenvalue 1 but every eigenvector with eigenvalue 1 need not be a distribution, i.e., it need not be a state vector.

Suppose the eigenvalues of M are in the range $[1, -1)$ and there is an eigenvector with eigenvalue 1. We noticed in the previous section: after repeated applications of M we will converge to this eigenvector with eigenvalue 1. The rate of convergence depends upon $1 - \mu$, where μ is eigenvalue with second highest absolute value.

Let us look at the consequence of this for the random walk on a graph and different state vectors u.

Exercise 6. Will the random walk, defined by $W = \frac{1}{d}A_G$ for a d-regular graph, converge?

Notice, because of Perron-Frobenius theorem, first eigenvector can be normalized to get a state vector. Hence, we have a stationary distribution.

We know that the eigenvalues of W lie in the range $[1, -1]$ and there is a stationary distribution. Though, there can be an eigenvector with eigenvalue -1. This happens if and only if the graph is bipartite.

Exercise 7. Can you show that the walk W will not converge for every state vector u on a bipartite graph with a direct argument?

The walk can be modified a bit to make it converge for every state vector. One way to do it is to make the walk *lazy*. A lazy random walk is one where with probability half we stay at the same vertex move to the neighbours with equal probability in the remaining half.

The lazy random walk operator is,

$$W_L = \frac{1}{2}(I + \frac{1}{d}A_G).$$

Exercise 8. What is the range of eigenvalues for the lazy walk operator W_L? What is the relation between eigenvalues of W and W_L?

Let $1 = \mu_1 \geq \mu_2 \geq \cdots \mu_n \geq 0$ be the eigenvalues of the lazy random walk operator. From the previous section, lazy random walk converges to the stationary distribution. Notice that the stationary distribution in this case is the state vector $\mathbf{1}$, the uniform distribution over all vertices. The walk converges to uniform distribution pretty fast; after $O(\frac{\log(n)}{1-\mu_2})$ steps it is $1 - 1/n$ close to the uniform distribution.

What does this mean for our $s - t$ connectivity problem? This means that after $O(\frac{\log(n)}{1-\mu_2})$ steps, the probability of reaching t is close to $1/n$. So, if we run this walk n times, with constant probability we will hit t.

1.3 Random walk on a cycle

Suppose the underlying graph is a cycle on n vertices. Let us analyze the stationary distribution and convergence of random walk on this graph.

The stationary distribution is the uniform distribution. For convergence, we need to find the second eigenvalue of $\frac{1}{2}(I + \frac{1}{2}C_n)$, where C_n is the adjacency matrix of the cycle on n vertices.

The eigenvalues of $\frac{1}{2}(I + \frac{1}{2}C_n)$ are $1/2 + 1/2\lambda_i$, where λ_i are the eigenvalues of $\frac{1}{2}C_n$. Let ω be the primitive n-th root of unity. You will show in the assignment that $(1, \omega^k, \omega^{2k}, \cdots, \omega^{(n-1)k})$, for all k, are the eigenvectors of C_n.

Exercise 9. Show that the eigenvalue for eigenvector corresponding to $(1, \omega^k, \omega^{2k}, \cdots, \omega^{(n-1)k})$ is $\omega^k + \omega^{n-k} = 2\cos(2\pi k/n)$.

So, the second eigenvalue for the lazy walk matrix is $\frac{1}{2}(1 + \cos(2\pi/n)) = \cos^2(\pi/n)$. Using the Taylor series expansion of cos function, $1 - \cos^2(\pi/n) = O(1/n^2)$. So, the random walk will converge to uniform distribution after $O(n^2 \log(n))$ steps.

2 Error reduction using random walk

We will look at an application of random walks in this section.

Suppose you are given a randomized algorithm which fails with probability $\frac{1}{3}$. We can repeat the algorithm a few times, take majority and decrease the error probability to be less than $\frac{1}{81}$.

Exercise 10. Calculate and convince yourself that if you repeat the algorithm constant number of times then the probability of failure is less than $\frac{1}{81}$.

Since this step will only take a constant overhead, let us assume that a standard randomized algorithm has a failure probability of at most $\frac{1}{81}$.

The problem is to reduce this error probability to something very small (say $\frac{1}{n}$ or $(\frac{1}{2})^n$). The most natural way would be to use the same trick we did above. Repeat the algorithm multiple times and take majority.

This strategy works and you will show that in the assignment. Though, this takes a lot of random bits. If we needed r random bits in every iteration and there were t iterations, we would need rt random bits.

Using random walk on graphs which expand well, we will show that we can only use $r + \log(d).t$ random bits to decrease the probability exponentially, where d is the degree of the graph.

2.1 Expanders

Remember that $W = \frac{1}{d}A_G$ is the walk matrix for a d-regular graph. We already saw that a random walk on the graph G reaches the stationary distribution soon if $|\mu_i|$ is small (for all $i \neq 1$). Here, μ_i are the eigenvalues of A_G.

A sequence of d-regular graphs (think of d as constant and increasing number of vertices) are called an expander family if $|\mu_i| \leq \epsilon d$ for all $i \neq 1$. This implies that the largest eigenvalue of the *walk matrix* is 1 and everything else has absolute value smaller than ϵ.

We will not worry about constructing such graphs. There is lot of literature on expanders, but we will only use the fact that there are constant degree expanders with $\epsilon = \frac{1}{10}$. Let us say, like before, that the algorithm uses r random bits in every iteration.

We assume an expander of constant degree, say d, with vertices indexed by $\{0,1\}^r$ and $\epsilon = \frac{1}{10}$. Notice that the number of vertices are huge, but we don't need the entire graph. Just an oracle is good if it allows us to move from a vertex to its neighbours in polynomial time. We will assume that such an oracle exists.

Let us instead focus on how these expanders can be used to reduce the amount of random bits needed in repeating our algorithm. We know that only $\frac{1}{81}$ fraction of the vertices are bad, i.e., they give the incorrect answer. Call this set of vertices F. The idea would be to take a random walk and show that we will mostly be outside F, since the graph is an expander. In other words, we will not be trapped in the set F for a long time (look at the figure below, Fig. 2.1).

So, to reduce the error, our strategy will be to start with a random vertex (it will take r random bits) and then take a random walk (every step will take only $\log(d)$ random bits). If we repeat this walk t times, we will use $r + \log(d).t$ random bits. In the next subsection, we will show that the error reduces exponentially in t.

2.2 Analysis of the error reduction in the random walk

Before we start the analysis, let us summarize the properties of d-regular graph on which we will take the random walk.

- The vertex set is $\{0,1\}^r$, where r is the number of random bits in the algorithm.
- The walk matrix is $\frac{1}{d}A_G$, and all the eigenvalues of W have absolute value less than $\frac{1}{10}$ (except the first one).
- There is a bad subset F, with at most $\frac{1}{81}$ fraction of total vertices, where our algorithm fails.

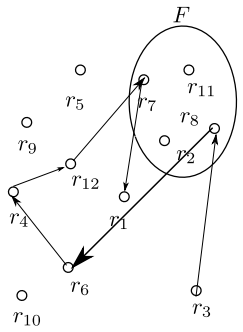

Fig. 1. Random walk on an expander

We want to show: if we take a random walk with t steps then the probability that at least half of the steps are in F, is exponentially low in t.

Our first choice is truly random, so the initial probability vector p_0 is $\frac{1}{n}\mathbf{1}$.

For a set $S \subseteq \{0,1\}^r$, let D_S be a diagonal matrix. We will set the i-th diagonal entry 1 if i is in S and zero otherwise.

Exercise 11. Show that the probability that we are in set F, given the state vector is p, is $\mathbf{1}^T D_F p$.

After one step of the walk, the state vector will be Wp_0 and the probability that we end up in F will be $\mathbf{1}^T D_F W p_0$.

Say, we are given a $\{0,1\}$ sequence s of length t which specifies the steps when the random walk was in F; if s_i is 1 then the random walk was in F at step i, otherwise, it was not in F. The probability that a random walk of t steps follows the sequence s is,

$$Prob(\text{random walk follows } s) = \mathbf{1}^T D_t W D_{t-1} W \cdots D_2 W D_1 W p_0.$$

Here, D_i is D_F if $s_i = 1$ and $I - D_F$ if $s_i = 0$.

Note 2. $Wp_0 = p_0$, so it does not matter if we take W at the first step or not.

Exercise 12. Convince yourself that the probability of being in sequence s is correct.

The quantity above is bounded by $\left\|\mathbf{1}^T\right\| \|D_t W\| \|D_{t-1} W\| \cdots \|D_2 W\| \|D_1 W\| \|p_0\|$. For the matrix norm, we use the spectral norm of a matrix. It is the maximum possible norm of Mv where v is a unit vector. We also use the result,

$$\|M_1 M_2\| \leq \|M_1\| \|M_2\|.$$

You will prove this in the assignment. You will also prove, for a symmetric matrix M, its norm $\|M\|$ is the highest absolute eigenvalue.

To bound the probability that random walk follows the sequence s, we need to bound the matrix norm $\|D_F W\|$.

Exercise 13. Why should the norm $\|D_F W\|$ be small?

Lemma 1.

$$\|D_F W\| \leq \frac{1}{9} + \frac{1}{10} \leq \frac{2}{9}.$$

Proof. We want to bound the norm of $D_F W x$, where x is a unit vector. Say, $x = c\mathbf{1} + y$ where y is perpendicular to $\mathbf{1}$. Then,

$$D_F W x = cD_F W\mathbf{1} + D_F W y = cD_F\mathbf{1} + D_F W y.$$

Using triangle inequality for the norm,

$$D_F W x \leq c\|D_F\mathbf{1}\| + \|D_F W y\|. \tag{1}$$

Exercise 14. Show that $c \leq \frac{1}{\sqrt{2^r}}$ because x is a unit vector. Hence, $c\|D_F\mathbf{1}\| \leq \frac{1}{9}$.

To bound the second term, notice that $\|D_F W y\| \leq \|W y\|$, since $\|D_F\| = 1$. We know that y is perpendicular to $\mathbf{1}$, and all eigenvalues of W have absolute value less than $\frac{1}{10}$ except for the eigenvector $\mathbf{1}$.

Exercise 15. Show that $\|W y\| \leq \frac{1}{10}$.

Using the bound on both terms of Eqn. 1, we get,

$$\|D_F W\| \leq \frac{1}{9} + \frac{1}{10} \leq \frac{2}{9}.$$

□

If any sequence s of random walk stays in F majority of times, then using Lem. 1,

$$Prob(\text{random walk follows } s) \leq \left\|\mathbf{1}^T\right\|\|D_t W\|\|D_{t-1} W\|\cdots\|D_2 W\|\|D_1 W\|\|p_0\| \leq \left(\frac{2}{9}\right)^{t/2}.$$

Here, $\|\mathbf{1}\| = \sqrt{2^r}$ and $\|p_0\| = \frac{1}{\sqrt{2^r}}$.

There are at most 2^t choices for s. So, the total probability that random walk is inside F in more than half the steps of random walk,

$$Prob(\text{majority of random walk in } F) \leq 2^t \left(\frac{2}{9}\right)^{t/2} \leq \left(\frac{8}{9}\right)^{t/2}.$$

So, we have shown that the probability of being in F for majority of time steps is exponentially small in t.

Hence, our algorithm will succeed with probability exponentially close to 1. Also, it will take only $r + \log(d).t$ random bits instead of rt random bits.

3 Properties of expander graph

We will look at some properties of expander graph in this section. This properties provide the intuition behind various applications of expander graph.

Let us remind ourselves the definition of an expander graph first. We will call a graph $G = (V, E)$ to be an α, d expander iff it is d-regular and

$$|\mu_i| \leq \alpha d,$$

for all $i \neq 1$. Where $\mu_1 \geq \mu_2 \cdots \geq \mu_n$ are the eigenvalues of the adjacency matrix A_G.

Strictly speaking, such graphs are called spectral expanders.

Exercise 16. Read the definition of combinatorial expander.

Spectral expansion implies combinatorial expansion by Cheeger's inequality. Combinatorial expansion implies spectral expansion can also be proved. In both these cases, the parameters need to be changed. We will only worry about spectral expansion in this course.

Before we go on to describe the properties of a spectral expander, let us introduce the concept of graph inequalities. Again, remember that we use the notation $A \succeq B$ iff $A - B$ is positive semidefinite. It is equivalent to saying that $x^T Ax \geq x^T Bx$ for all x.

Exercise 17. Show that it is enough to consider all unit vectors x.

We will introduce the notation $G \succeq H$ iff $L_G \succeq L_H$.

Exercise 18. Let H be a sub-graph of G, show that,

$$G \succeq H.$$

Let us take a look at an example where such kind of inequalities are proved. Let $G_{1,n}$ be the graph with n vertices and just one edge, from 1 to n. Let P_n be a path on n vertices where vertex i is connected to vertex $i+1$. These two graphs are related by the inequality,

$$(n-1)P_n \succeq G_{1,n}.$$

For this inequality, we need to prove,

$$(n-1)\sum_{i=1}^{n-1}(x_i - x_{i+1})^2 \geq (x_1 - x_n)^2.$$

This follows by the following observation,

$$(x_1 - x_n)^2 = \left(\sum_{i=1}^{n-1} x_i - x_{i+1}\right)^2 \leq (n-1)\sum_{i=1}^{n-1}(x_i - x_{i+1})^2.$$

Where, last inequality follows from Cauchy-Schwartz.

3.1 Approximation of complete graph

A graph H is called an ϵ *approximation* of G iff

$$(1-\epsilon)H \preceq G \preceq (1+\epsilon)H.$$

With the notion of graph approximation, we will now show that expanders are sparse approximations of a complete graph. A complete graph on n vertices, K_n, has $\frac{n(n-1)}{2}$ edges. A d-regular expander on n vertices has $\frac{dn}{2}$ edges.

We know that expanders exist with constant degree. The following theorem shows that a complete graph with $\theta(n^2)$ edges can be approximated by a graph with only $O(n)$ edges.

Theorem 1. *Define H to be the graph $\frac{d}{n}K_n$, i.e., the Laplacian of H is the matrix $\frac{d}{n}L_{K_n}$. Let G be an α, d expander, then*

$$(1-\alpha)H \preceq G \preceq (1+\alpha)H.$$

Proof. Let us start with the simple observation.

Exercise 19. If $x = \pm\mathbf{1}$, show that,

$$L_H x = L_G x = 0.$$

Any vector x can be written as $c_1\mathbf{1} + y$ where $y \perp \mathbf{1}$. From the previous exercise,

$$x^T L_H x = y^T L_H y \text{ and } x^T L_G x = y^T L_G y.$$

This shows that we can assume x to be orthogonal to $\mathbf{1}$. So, we only need to prove, for any $x \perp \mathbf{1}$,

$$(1-\alpha)x^T L_H x \preceq x^T L_G x \preceq (1+\alpha)x^T L_H x.$$

We know that all the eigenvalues of A_G are less than αd in absolute value except the first one. Translating this statement to Laplacian,

$$|d - \lambda_i| \leq \alpha d \text{ for all } i \neq 1.$$

In other words, all eigenvalues λ_i, for i not equal to 1, lie in the range $(d - \alpha d, d + \alpha d)$.

We know that if $x \perp \mathbf{1}$, then it lies in the span of eigenvectors excluding the first eigenvector ($\mathbf{1}$). So, for a unit vector x,

$$(d - \alpha d) \leq x^T L_G x \leq (d + \alpha d).$$

This implies,

$$(1-\alpha)dx^T x \leq x^T L_G x \leq (1+\alpha)dx^T x,$$

since x is a unit vector.

Notice that $K_n = nI - J$, where J is the all 1's matrix. So, we get that $x^T L_H x = \frac{d}{n} x^T nIx = dx^T x$, for all $x \perp \mathbf{1}$. This proves that,

$$(1-\alpha)x^T L_H x \preceq x^T L_G x \preceq (1+\alpha)x^T L_H x.$$

□

3.2 Expanders as random graphs

Cheeger's inequality shows that if λ_2 is high then the graph has high expansion. In other words, the number of edges between S and $\bar{S}$ are high.

For a graph to be an expander, it needs to have high λ_2 but also λ_n is not far away from d. With these stricter conditions, we can show that the number of edges between any two sets S and T in an expander behaves like the same quantity in a random graph.

If $|S| = \beta n$, then the probability that a random u is in S is β. If $|T| = \gamma n$, then the probability that a random ordered pair (u, v) has $u \in S$ and $v \in T$ is $\beta\gamma$.

So, we expect to have $\beta\gamma(2E)$ edges between S and T in a random graph. Factor of 2 occurs because we are considering ordered pairs and E is the number of total edges.

We will show that the number of edges between S and T in an expander is close to $\beta\gamma dn$, like a random graph. To prove this result, we will only use the fact that an expander is close to a complete graph, Thm. 1.

Theorem 2. *Let H α approximate G, where $G = (V, E)$ is a d-regular graph and $H = \frac{d}{n}K_n$. If $S \subseteq V$ has βn vertices and $T \subseteq V$ has γn vertices then,*

$$|E(S,T) - \beta\gamma dn| \leq \alpha dn\sqrt{\beta\gamma}.$$

Note 3. The quantity $E(S, T)$ counts the ordered pair of edges between S and T. In other words, if u, v are both in S as well as T, we count edge u, v twice.

Proof. Let x_S denote the indicator vector for subset S and similarly x_T for T. We are interested in the quantity $x_S^T(L_H - L_G)x_T$. To see why, notice that

$$x_S^T L_G x_T = \sum_{i,j}(L_G)_{i,j}(x_S)_i(x_T)_j = \sum_i d(x_S)_i(x_T)_i - \sum_{(i,j)\in E_G}(x_S)_i(x_T)_j = d|S \cap T| - E(S,T).$$

Since $H = \frac{d}{n}(nI - J) = dI - \frac{d}{n}J$, we can say,

$$x_S^T L_H x_T = x_S^T (dI - \frac{d}{n}J)x_T = dx_S^T x_T - \frac{d}{n}\sum_{i,j}(x_S)_i(x_T)_j = d|S \cap T| - \beta\gamma dn.$$

So, $x_S^T(L_H - L_G)x_T = E(S,T) - \beta\gamma dn$. So, we only need to prove that $\left|x_S^T(L_H - L_G)x_T\right| \leq \alpha dn\sqrt{\beta\gamma}$.

We will bound the left hand side using the fact that $\|L_H - L_G\|$ is small. You will prove in the assignment that since H approximates G, $\|L_H - L_G\| \leq \alpha d$.

The following chain of inequalities will complete the proof.

$$\begin{aligned}\left|x_S^T(L_H - L_G)x_T\right| &\leq \|L_H - L_G\|\|x_S\|\|x_T\| \\ &\leq \alpha d\|x_S\|\|x_T\| \\ &\leq \alpha dn\sqrt{\beta\gamma}.\end{aligned}$$

Here, the last inequality follows from the following exercise.

Exercise 20. Prove that $|x_S|^2 = \beta n$ and $|x_T|^2 = \gamma n$.

□

Using Thm. 2, we can show that the neighbourhood $N(S)$ of any set $S \subseteq V$ is big.

Let $|S| = \beta n$ and $|N(S)| = \gamma n$. This implies that $|V - N(S)| = (1-\gamma)n$. Applying Thm. 2 on S and $V - N(S)$,

$$|E(S, V - N(S)) - \beta(1-\gamma)dn| \leq \alpha dn\sqrt{\beta(1-\gamma)}.$$

Since there are no edges between S and $V - N(S)$,

$$\beta(1-\gamma)dn \leq \alpha dn\sqrt{\beta(1-\gamma)}.$$

Simplifying,

$$\sqrt{\beta(1-\gamma)} \leq \alpha.$$

Squaring both sides and simplifying,

$$\gamma \geq 1 - \frac{\alpha^2}{\beta}.$$

This shows that if G is an α, d expander and S is a set with size βn, then the neighbourhood of S has size at least $(1 - \frac{\alpha^2}{\beta})n$.

4 Assignment

Exercise 21. Read about Markov chains and their connection with random walks.

Exercise 22. Show that the lazy walk operator for a general graph is $\frac{1}{2}(I + A_G D_G^{-1})$. Also, show that it is similar to normalized Laplacian.

Exercise 23. Show that a lazy walk on a general graph converges using the walk operator given in the previous question.

Exercise 24. Analyze the stationary distribution and convergence of lazy random walk on complete graph K_n.

Exercise 25. Find the definition of circulant matrices? Show that all the eigenvectors of a circulant matrix are $(1, \omega^k, \omega_2^k, \cdots, \omega_{n-1}^k)$, for all k.

Exercise 26. Show that the error probability decreases exponentially when we repeat a randomized algorithm and take majority (hint: use Chernoff bound).

Exercise 27. Read about spectral norm. Show that for a symmetric matrix, it is the highest absolute eigenvalue. Prove that,

$$\|M_1 M_2\| \leq \|M_1\| \|M_2\|.$$

Exercise 28. Show that if H is an ϵ approximation of G, then

$$(1-\epsilon)\lambda_k(H) \preceq \lambda_k(G) \preceq (1+\epsilon)\lambda_k(H),$$

for all k.

Exercise 29. Given L_H is an ϵ approximation of L_G, show that,

$$\|L_G - L_H\| \leq \epsilon d.$$

Where $H = \frac{d}{n} K_n$, G is a d-regular graph and norm denotes the spectral norm.