Expander graph is an ubiquitous mathematical object in graph theory. It has various applications in diverse disciplines including applied math, computer science, geometry, probability etc. Discovered in 1970s and being proved existing by Pinsker, expander graph is indeed on of the most important mathematical invention.

Intuitively, expander graph is a family of **sparse** but **highly connected** graphs, or, an approximation of complete graph. One can view it with three different aspects: combinatorially/geometrically, probabilistically, and algebraically.

- Combinatorially/geometrically: Directly from its definition, any not too large subset of an expander graph has many neighbors. That is, one can not find a subset of expander graph that can be easily cut out from others.
- Probabilistic: When performing a random walk on an expander graph, the convergence rate to stationary distribution is very fast. Moreover, since it's sparse (the degree of each vertex is constant), a random walk does not require to many randomness. As a result, it is a powerful tools for pseudorandomness and applications in Markov chain etc.
- Algebraically: The extent of expansion is also related to the spectrum of a graph, i.e. the second largest eigenvalue of the normalized adjacency matrix of the graph. As lone as the expansion is large, the second eigenvalue will be bounded away from 1, which is the largest eigenvalue. In this sense, we have a computationally efficient way to compute the level of expansion.

With the above three different point of views, expander graphs have various definition and each of them has relation to others with certain parameters setting.

Basically, expander graphs have the following research fields that are still received high attention:

- How to explicitly construct good expander graphs? Currently the best explicit construction is the so called Ramanujan graph, which, however, cannot achieve the expansion level from probabilistic argument. (In fact, there's a 1/2 factor) As a result, people are still looking forward for an efficient and optimal construction for expander graphs.
- Relation among different definitions. As we discussed above, different definition of expansion has different practical/theoretical implications. There are lots of known results proving some parameter transformation among distinct definitions, which will be introduced in later posts. However, you will see that there exists gap between these transformation. Thus, to show the optimality or impossibility is also an important line of works.

In this series of posts, I will summarize important results about expander graph and some of my reflections. The materials I studied is the Pseudorandomness lecture notes by professor Salil Vadhan from Harvard university and the nice survey paper by professor Hoory, Linial, and Wigderson.

Motivations

Here I will show a beautiful application of expander graph: error reduction in RP. There are many others nice results such as error-correcting codes, circuit lower bound etc. One can find nice resources from the two materials I mentioned.

To start with, let me formulate the problem. As we know from previous introduction about randomized computation, RP is a complexity class that consists of problems having polynomial time algorithm with no false positive error while allowing false negative error. Since for the convenience of definition and construction, we only apply a loose bound 1/2 for the false negative error. What if we want the error to be negligible, say 2^{-k}?

The simplest way is to repeat the algorithm k times so that by Chernoff bound, we can reduce the error to 2^{-k}. However, suppose the randomness required for single algorithm is m bits, then after this trivial reduction, the required randomness will blow up to mk bits.

Another approach is to use pairwise-independent randomness. The randomness can be reduced to $O(m+k)$, however, the running time will increase exponentially. Surprisingly, with expander graphs, doing random walk on it to generate pseudorandom bits with only constant randomness each round can successfully reduce the randomness while in the meantime preserve running time in constant factor. The results is compared in the following table.

	Running Time	Randomness
Trivial	$O(k)$	$O(mk)$
Pairwise-Independence	$O(2^k)$	$O(m+k)$
Expander graph	$O(k)$	$m+O(k)$

Details of this error reduction will be discussed in later posts after we formally define the notion of expansion and deriving several useful theorems and lemmas. For now, readers can have a taste of the power of expander graph.

In the previous post (Expander Graphs - Introduction and Motivation), we said that expander graphs is a family of **sparse** but **highly connected** graphs, which are approximations for complete graphs. Now, we are going to elaborate on how to properly define the notion of sparse and highly connected properties of a graph so that we can evaluate how well a expander graph is. We call this evaluation for expander graphs as the measure of expansion.

As a matter of fact, there are many definitions for the measure of expansion. In this post, I'm going to focus on the three fundamental one: vertex expansion, spectral expansion, and edge expansion. These three definitions capture different aspects of expander graphs: Vertex expansion considers the number of neighbors of any not too large subset of vertices, which directly captures the highly connected property of graphs. Spectral expansion considers the second largest eigenvalue of random walk matrix, which captures the speed of convergence. Edge expansion considers the number of edges among subset of vertices and its complement, which captures the probability of leaving this subset.

Although these three measures of expansion have different definitions, since they all want to interpret the notion of sparse but highly connected property, there are certain equivalent relation among them under specific parameters setting. In the rest part of this post, I will first formally introduce the definition of these three measures, then present some equivalent relation. Finally, I will make a summary table and discuss the implication of the parameters setting in the equivalent relation.

Vertex expansion

Intuitively, vertex expansion captures the graphs with the property that for any not too large subset of vertices, the number of their neighbors is not too small. Formally, we have

Definition 1 (Vertex expansion) *G is a (K, A) vertex expansion if $\forall S \subseteq V(G)$, $|S| \leq K$, $|N(S)| \geq A \cdot |S|$.*

Here, we denote the neighbor of S as $N(S) := \{u : u \in V(G) \backslash S,\ v \in S,\ (u, v) \in E(G)\}$. That is, the vertices that have an edge connect to S. Similarly, we could define the vertex expansion with a different neighbor referring to the vertices that have edge from S. Basically, this doesn't affect our reasoning.

Most of the time we will focus on $D-$regular graphs. And we can immediately see that for any subset size regulation, we cannot achieve arbitrary neighbor factor A. Formally, we have the following lemma about the upper bound for A in a regular graph.

Lemma 2 (Upper bound for A) *Let G be a $D-$regular graph with N vertices. Suppose G is a $(\alpha N, A)$ expander for some $\alpha > 0$, then $A \leq D - 1 + O(1)$, where the $O(1)$ factor vanishes as $N \to \infty$.*

I proved this upper bound in the Problem 3 of Problem Set 4 of Harvard CS225.

After knowing the existence of an upper bound, we would like to find out how tight it is and try to attack it with some explicit construction. First of all, with probabilistic setting, we could see that a random $D-$regular graph will be an $(\alpha N, D - 1.01)$ vertex expander with high probability. However, the best explicit construction of $D-$regular graph can only achieve $(\alpha N, D/2)$ vertex expander.

Spectral expansion

Intuitively, spectral expansion wants to use the difference of the second largest eigenvalue λ of random walk matrix and 1 to capture the speed of convergence of a random walk to its stationary distribution. As long as λ is far from 1, the convergence speed will be very fast. Thus, we define the spectral expansion as follow:

Definition 3 (Spectral expansion) *Let $\lambda(R)$ to be the second largest eigenvalue of R, which is the random walk matrix of graph G. If $\gamma(R) = 1 - \lambda(R) \geq \gamma$, then we say G has spectral expansion γ.*

For a $D-$regular graph, it's easy to see that we cannot achieve arbitrary spectral expansion. Namely, there's an upper bound for spectral expansion. Specifically, we have

Lemma 4 (Upper bound for spectral expansion) *$\forall$ $D-$regular multigraph G with N vertices, we have $\lambda(G) \geq \frac{2\sqrt{D-1}}{D} - O(1)$, i.e. $\gamma \leq 1 - \frac{2\sqrt{D-1}}{D} + O(1)$, where $O(1)$ vanishes to 0 as $N \to \infty$.*

I proved this upper bound in the Problem 4 of Problem Set 4 of Harvard CS225.

Although having an undesired upper bound for spectral expansion, a good news is that there exists a family of graph, Ramanujan graph, which asymptotically achieve the bound. Details about Ramanujan graph will be discussed in latter post.

Edge expansion

Edge expansion considers the number of edges among a subset of vertices and its complement. The formal definition is as follow.

Definition 5 (Edge expansion) *We say a $D-$regular graph G is a (K, ϵ) edge expander if $\forall S \subseteq V(G)$ with $|S| \leq K$, $e(S, \bar{S}) \geq \epsilon \cdot |S| \cdot D$.*

Intuitively, the definition of edge expansion is equivalent to asking $\frac{e(S,\bar{S})}{|S| \cdot D} \geq \epsilon$ for any not too large vertex set S, while the ratio $\frac{e(S,\bar{S})}{|S| \cdot D}$ can be thought of as the probability that, if we conditioned the stationary distribution on S, the random walk leaves S in a single step.

Equivalent relation among distinct measures

For simplicity, here we only consider two relations: vertex expansion versus spectral expansion, and edge expansion versus spectral expansion.

- Vertex expansion versus Spectral expansion

Relation	Vertex expansion
$\Rightarrow$	G is a $D-$regular $(\frac{N}{2}, 1+\delta)$ vertex expander.
$\Leftarrow$	G is an $(\frac{N}{2}, 1+\gamma)$vertex expander. More generally, a $(\alpha N, \frac{1}{(1-\alpha)}$
$\Leftrightarrow$(G is a $D-$regulr multigraph)	$\exists \delta > 0$ such that G is an $(\frac{N}{2}, 1+\delta)$ vertex expander.

- Edge expansion versus Spectral expansion

Relation	Edge expansion	Spectral expansion
$\Rightarrow$	G is an $(\frac{N}{2}, \epsilon)$ edge expander.	Let α be the portion of self-loops for all vertices in G, t
$\Leftarrow$	G is an $(\frac{N}{2}, \frac{\gamma}{2})$ edge expander.	G is a regular digraph with spectral expansion γ.

From the above the equivalent relation among different measures of expansion, we can see that each definition is not exactly the same. That is, there are still some constant/small factor affecting the transformation between two measures. The fundamental reason might lie in the differences of their focusing, e.g. spectral expansion look at a more global property of a graph, while the other two adopt local and combinatorial definitions. As a result, graphs having the same vertex expansion might behave unbalanced in a global sense and resulting in a factor lost, e.g. the $1/D^2$ lost during transformation from vertex expansion to spectral expansion.

Different definitions of expansion have distinct properties. It depends on the applications to decide which one is better. In the following posts, I will introduce some nice results, e.g. Expander mixing lemma, so that we can see how to make good use the idea of expander graphs.

For computer scientist, what we care the most about expander graphs is performing random walks on them. As what we discussed in the first post, we can think of an expander as approximation of complete graph. That is, it provides randomness to certain level so that we can spend less fully randomness while in the meantime have nice performance.

Concretely, this is a tradeoff among the number of random bits, computing time, number of samples, and error. Using less randomness will increase the complexity of time and error. However, if we can strike a balance between these factors, it is possible to achieve lower randomness level without blowing the time to exponential level.

As a result, in this post, our goal is to analyze the properties of random walks on expanders so that we can see how do we trade error and time with the reduction of randomness. The central idea is to upper bound the distance between the distribution after t steps and the uniform distribution. Not too surprisingly, the spectral expansion plays an important role in our analysis. Intuitively, as we have spectral expansion $1-\lambda$, when λ is at our desired error level, then we can successfully reduce the number of random bits.

In the following technical parts, we will start from introducing two basic analyzing tools of expander graphs: vector decomposition and matrix decomposition. Then, we derived two important theorems: hitting theorem and Chernoff bound for expander. With these two theorems, we can finally have randomness-efficient algorithms for error reduction and sampling.

Two approaches to analyze random walks

When we randomly walk on an expander, we would like to know the evolution of distribution. Consider the simplest case: starting from distribution v, what's the distribution after one random step? How far is it to the uniform distribution? Assume the random walk matrix of the expander is M, then our goal is to analyze the behavior of vM.

There are two basic approaches to do so: vector decomposition and matrix decomposition. The former rewrite the distribution into two parts, one being parallel to the uniform distribution and another being orthogonal to the uniform distribution. The latter rewrite the random walk matrix into two parts, one being a complete graph and another being a matrix with small eigenvalues. After the decomposition, people can then play with each part respectively and most of the time this will simplify the analysis. The following describe the decompositions in a more rigorous way.

- Vector decomposition
  Given input distribution v, decompose it as $v = v^{\parallel} + v^{\perp}$, where $v^{\parallel} = \frac{<v,u>}{<u,u>}$

is parallel to the uniform distribution u and $v^{\perp} = v - v^{\parallel}$ is orthogonal to u. Observe that since u remains u after one step, we have

  $vM = (v^{\parallel} + v^{\perp})M = v^{\parallel} + v^{\perp}M$

  Suppose the spectral expansion of M is $1 - \lambda$, we have

  $||vM||_2^2 = ||v^{\parallel}||_2^2 + ||v^{\perp}M||_2^2 \leq ||v^{\parallel}||_2^2 + \lambda||v^{\perp}||_2^2$

  In some applications, e.g. the expander construction or expander mixing lemma, the spectral expansion of M might be unknown, and we only know the expansion of certain part of M. Using vector decomposition can help us upper bound the λ of M when we can decompose the outcome vM. However, one should know that most of the it is not the case.

- Matrix decomposition
  Another approach is rewrite the random walk matrix M as

  $vM = \gamma v^{\parallel} + \lambda v^{\parallel} + v^{\perp}M = \gamma \cdot vJ + \lambda \cdot vE = v(\gamma \cdot J + \lambda \cdot E)$

  , where J is the complete graph and E is the error matrix satisfying $||vE||_2 \leq ||v||_2$.

With matrix decomposition, we can upper bound the λ of complicated random walk matrix. However, although it's more easy to analyze the spectral expansion with matrix decomposition than vector decomposition, the bound is usually looser.

Basically, the two decompositions have different intuitions, vector decomposition emphasize on uniform distribution while matrix decomposition focus on complete graph. As a result, people should choose the one which matches the underlying structure of your random walk matrix.

Two important theorems

When using expander graphs to generate pseudorandomness, we may concern the following two issues:

- The probability of staying in certain set of vertices.
- The difference between uniform distribution.

Once we can estimate the probability of staying in certain set of vertices, we can bound the error explicitly. To apply the pseudorandomness of expander graph, e.g. sampling average, we can bound the error with the difference to uniform distribution.

The probability of staying in certain set of vertices

For an expander with N vertices and spectral expansion λ, the probability of staying in a vertex set B after t steps from uniform distribution u is $(\mu + (1-\mu)\cdot\lambda)^t$, where $\mu = \frac{|B|}{N}$.

Here, we sketch the proof with matrix decomposition method.

Let P be the indication matrix for B where $P_{ii} = \mathbf{1}_{\{i \in B\}}$. Observe that

- The probability of being in B at the first step: $|uP|_1$.
- The probability of staying in B after t steps $|uP(MP)^{t-1}|_1 = |uP(PMP)^{t-1}|_1$.

From previous section, we know that we can decompose the random walk matrix as $M = (1-\lambda)\cdot J + \lambda E$, where $||E||_2 \leq 1$. We can then rewrite the error probability as $|uP(PMP)^{t-1}|_1 = |uP[P((1-\lambda)\cdot J + \lambda E)P]^{t-1}|_1 = |uP[(1-\lambda)\cdot PJP + \lambda \cdot PEP]^{t-1}|_1$. We have

- From 1-norm to 2-norm: $|uP[(1-\lambda)\cdot PJP + \lambda\cdot PEP]^{t-1}|_1 \leq \sqrt{\mu N}\cdot ||uP[(1-\lambda)\cdot PJP + \lambda\cdot PEP]^{t-1}||_2$.
- By Cauchy-Schwartz: $||uP[(1-\lambda)\cdot PJP + \lambda\cdot PEP]^{t-1}||_2 \leq ||uP||_2 \cdot |||[(1-\lambda)\cdot PJP + \lambda\cdot PEP]|||_2^{t-1}$.
- Upper bound for $||PEP||_2$: $||PEP||_2 \leq 1$.
- Upper bound for $||PJP||_2$: $||xPJP||_2 = (\sum_i (xP)_i^2 \cdot \mu^2)^{1/2} \leq \mu \cdot (\sum_i x_i^2)^{1/2} = \mu \cdot ||x||_2$.

Combine all the inequalities above, we have an upper bound for the error probability:

$|uP(PMP)^{t-1}|_1 \leq [(1-\lambda)\cdot\mu + \lambda]^{t-1} \leq [\mu + \lambda\cdot(1-\mu)]^t$

As $\mu < 1$, we can see that the error probability can converge to 0 exponentially fast.

The difference between uniform distribution

From the first theorem about random walk on expander graph, we can estimate the error when we exactly know what kind of random strings induce error. However, sometime we might consider error in different form. For instance, using sample average to estimate the expectation of a bounded function. The error in this problem is not a yes/no setting, instead, we would like to estimate the how wrong did we make. As a result, we need a theorem analogous to Chernoff bound for random walk on expander graph. The following theorem successfully presents a nice bound for us.

Theorem 1 (Chernoff bound for expander) *Given G an expander with spectral expansion $1-\lambda$ and N vertices. Let $V_1, \dots, V_t$ denotes the vertices it traverses, and function $f : [N] \to [0,1]$. We have*
$P[|\frac{1}{N}\sum_i f(V_i) - \mu(f)| \geq \epsilon] \geq \lambda + \epsilon \leq 2e^{-\Omega(\epsilon^2 t)}$.

Once we take $\lambda = \epsilon$, we can yield a nice upper bound for error.

To prove the theorem, one can follow the idea of original Chernoff bound to use moment generating function and then use matrix decomposition to tune the parameter up to desired level.

Applications of random walk on expanders

Basically, we can use expander to generate pseudorandomness and reduce the number of full random bits required. In this section, I will present two fundamental examples which show the power of expander. For each example, we can solve it in three ways: truly independent randomness, pairwise randomness, and random walk on expander. We will compare the parameters of them to achieve the same error level.

Error reduction

Suppose there's a **BPP** algorithm using m random bits to yield constant error 1/3. To reduce the error to exponentially small, say 2^{-k}, we need to repeat the algorithm for t times and use r random bits as follow:

Method	Number of repetitions	Number of random bits
Fully randomness	$O(k)$ $O(mk)$	
Pairwise independence	$O(2^k)$	$O(m+k)$
Expander walks	$O(k)$	$m + O(k)$

Sampling

Suppose there's a function $f : [N] \to [0,1]$, the number of samples, and random bits to approximate the mean of f with with probability $1-\delta$ and additive error less than ϵ are:

Method	Number of repetitions	Number of random bits
Fully randomness	$O(\log(1/\epsilon^2) \cdot \log(1/\delta))$ $O(m \cdot (1/\epsilon^2) \cdot \log(1/\delta))$	
Pairwise independence	$O(1/\epsilon^2 \cdot 1/\delta)$	$O(m + \log(1/\epsilon) + \log(1/\delta))$
Expander walks	$O((1/\epsilon^2) \cdot \log(1/\delta))$	$m + O(\log(1/\delta) \cdot (\log(1/\epsilon))/\epsilon^2)$

One can see that there's no dominant method here since expander walk performs better in the number of samples while pairwise independence performs better in the number of random bits.

In this post, we are going to talk about the construction of expander graphs. First of all, let's have a discussion on why should we care about the construction of expanders and what's our goals.

Motivations and Goals

From the previous posts (Introduction and Motivation, Random Walks), we can see that expander has nice applications in reducing the amount of randomness required. However, in Measure of expansions, we also discovered that we cannot achieve arbitrary degree of expansion. Moreover, only from probabilistic argument can we show the existence of nice expander. That is to say, we need to generate an expander randomly in order to reduce the randomness in its applications! As a result, it's tempting to find **explicit** approaches to construct a nice expander.

Before we start introducing the explicit construction for expander, let's think about what do we mean by a *nice* expander? First, we would like to have a graph with expansion as large as possible, at least with a constant far away from zero. Next, we also hope the degree of the graph could be small, more precisely, bounded by a constant, so that in each step of random walk, the randomness required is also constant. Finally, we would like to have a family of nice expanders with arbitrary number of vertices so that we can deal with problem with any size. Clearly that it is difficult for us to do so for every possible number of vertices. However, realistically, we would at least want to construct a family of expanders with a slow growing number of vertices such that it's convenient for us to use.

To summarize, our goals are:

- Large expansion. (Bounded away from 0)
- Small degree. (Bounded above by a constant)
- The number of vertices grows slowly. (at most polynomially fast)

Graph Operations

From the previous section, we have three goals for constructing a nice expander. However, at first glimpse, we might discover that it is extremely non-trivial to achieve these three criterion all at once. For instance, as we want to increase the expansion of the graph, the degree will also grow, and vice versa. That is, to construct a nice expander in one step seems to be a tough job.

As a result, some smart people came up with an idea to construct a nice expander step by step. Concretely, in some steps, we can increase the expansion while in the meantime increase a little bit degree. In other steps, we decrease the degree while in the meantime do not hurt too much on the expansion. As a matter of fact, we have the following three **graph operations**, which are served with different effects. After some compositions and analysis, we can construct a nice expander explicitly. In the following table, let's see how these three graph operations work.

Graph Operation	Before	After	Intuition
Squaring	$G : (N, D, \gamma)$	$G^2 : (N, D^2, 2\gamma - \gamma^2)$	Increase tl
Tensoring	$G_1 : (N_1, D_1, \gamma_1) G_2 : (N_2, D_2, \gamma_2)$	$G_1 \otimes G_2 : (N_1 N_2, D_1, D_2, \min\{\gamma_1, \gamma_2\})$	Increase tl
Zig-zag product	$G : (N_1, D_1, \gamma_1) H : (D_1, D_2, \gamma_2)$	$G \textcircled{z} H : (N_1 D_1, D_2^2, \gamma_1 \gamma_2^2)$	Decrease t

Observe the above table, if we can start with a small expander with constant degree and constant expansion, then as we apply squaring and zig-zag product iteratively in a proper way, we can increase the size of expander polynomially without affecting the degree and expansion too much. That is, yielding a family of expander with constant degree and constant expansion.

We will formalize this idea in latter section, for now, let's have a deeper look at these graph operations so that we can have a feeling about the tradeoff among them.

Squaring

Given a graph $G = (V, E)$, we define its squaring as $G^2 = (V, E')$, where $E' = E \cup \{(i, k) | (i, j) \in E, (j, k) \in E\}$. That is, we turn a two steps edges in G into a new edge in G^2. Intuitively, we can think of as increasing the neighborhood of a graph in order to improve the expansion.

Tensoring

Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, we define their tensor product as $G_1 \otimes G_2 = (V', E')$, where $V' = V_1 \times V_2$ and $E' = \{((i, u), (j, v)) | (i, j) \in E_1, (u, v) \in E_2\}$.

Zig-zag product

Zig-zag product is a little bit more cumbersome than the previous two graph operations, however, it is the critical part in the construction of expander as it provides a way to lower the degree of a graph while in the meantime preserving the expansion.

To understand zig-zag product, it's better to start with a similar but simpler graph operation to get some geometric intuition: replacement product. Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$, where G is a $N-$vertices $D-$regular graph and H is a $D-$vertices $d-$regular graph. We define the replacement product of G and H as $G_r =$

$G\textcircled{r}H = (V_r, E_r)$ where $V_r = V_G \times V_H$ and $E_r = \{((u,i),(v,j))|v$ is the ith node of u, u is the jth node of $v\}$ $\{((u,i),(u,j))|(i,j) \in E_H\}$. Intuitively, we can think of the vertices in G_r from the same vertex in G as a cloud, where the relation of each cloud is exactly the same as the relation of the vertices in G. What we have done is extending each vertex in G into a cloud such that the degree of each vertex becomes $d+1$ while in the meantime the expansion of the graph is preserved. (This will be proved latter)

However, one can see that replacement product is asymmetric, that is to say, the existence of an edge from (u,i) to (v,j) does not guarantee the existence of edge from (v,j) to (u,i). To fix this issue, people came up with the brilliant idea: zig-zag product.

Here, we first formally state the definition of zig-zag product then show the geometrical intuition.

Similarly, we consider two graphs G and H as above and define the zig-zag product of G, H as $G_z = G\textcircled{z}H = (V_z, E_z)$, where $V_z = V_G \times V_H (= V_r)$ and $E_z = \{((u,i),(v,l)|\exists j,k$ s.t. $((u,i),(u,j)),((u,j),(,v,k)),((v,k),(v,l)) \in E_r\}$. That is, there's an edge from (u,i) to (v,l) iff we can go to (v,l) in three steps (in G_r):

1. Go to (u,j), which is in the same cloud as u.
2. Go to (v,k), which is in another cloud.
3. Go to (v,l), which is in the same cloud as v.

Note that this will end up with an $(ND, d^2, \gamma_1\gamma_2^2)$ expander, where γ_1 and γ_2 are expansion of Gm H respectively. (The resulting expansion is non-trivial to derive, which will be discussed later)

Expander construction

In this section, we are going to construct a nice expander family explicitly. Before we state and analyze the algorithm, let's take a look at the what do we mean by having a good construction.

Recall what we have discussed in the first section, our goal is to construct a family of expanders that have large expansion and small degree. Now, we also need to consider the time complexity issue, namely, how fast can we construct the desired expander? Here, we define two notions for a nice explicit expander construction.

Definition 1 (mildly explicit) *A construction is said to be mildly explicit if we can construct a complete representation of the expander in time poly(N), where N is the size of the expander.*

Definition 2 (fully explicit) *A construction is said to be fully explicit if given a node $u \in [N]$ and $i \in [D]$, we can compute the ith neighbor of u in time poly$(\log N)$, where D is the degree of the expander.*

For some applications, mildly explicit construction is enough while in some space-constrained applications, we require fully explicit construction.

In the following, we will present one mildly explicit construction with squaring and zig-zag product and one fully explicit construction with squaring, zig-zag product and tensoring. All of our following construction will be based on a constant-sized expander $H = (D^4, D, 7/8)$, where D is some constant. (proof of the existence of H can be found in Problem 4.8 of Problem Set 4 of Harvard CS225)

Mildly explicit construction

We construct a family of graph as follow:

1. Take $G_1 = H^2$.
2. Let $G_{t+1} = G_t^2 \textcircled{z} H$.

With some analysis, we will find that G_t is a $(D^{4t}, D^2, 1/2)$ expander. As a remark, the time complexity for constructing G_t is $N_t^{\Theta(1)}$.

Fully explicit construction

We construct a family of graph as follow:

1. Take $G_1 = H^2$.
2. Let $G_{t+1} = (G_t \otimes G_t)^2 \textcircled{z} H$.

With this construction, G_t is a $(C^{2^t}, D^2, 1/2)$ expander, where C is some constant. Since the exponentially growing size is not satisfactory, with a little modification, we can have G_t to be a $(D^{4t}, D^2, 1/2)$ expander. Notice that the time complexity of querying singe vertex is *poly*$(\log N_t)$.

Reflection

The explicit construction of expander graph is indeed a beautiful masterpiece in both applied math and TCS. The intuitive graph operations not only provide us a primitive to make good use of expander but also give insight into the properties of expander.