

Reducing randomness
w/ random walks

Lecture 1

Probability

Events, probabilities,
conditional probabilities,
union bound

Random variables,
expected value,
linearity of expectation
Markov's inequality

Algorithms

Approx. 3-SAT,

Big idea: we only want the average

Linearity of Expectation
 $E[X+Y] = E[X] + E[Y]$ (stochastically additive)
"average sum = sum of averages"

let $Y_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{otherwise} \end{cases}$ ($i=1, \dots, m$)

$E[Y_i] = 7/8$ (1 clause case)

$E[\text{\# clauses satisfied}] = E[\sum_{i=1}^m Y_i] = \sum_{i=1}^m E[Y_i] = 7/8 m$

let $Y_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{otherwise} \end{cases}$

$Z = \sum_{i=1}^m Y_i = \text{\# clauses satisfied}$

$E[Z] = \frac{1}{2} E[Z | x_1 = \text{true}] + \frac{1}{2} E[Z | x_1 = \text{false}]$

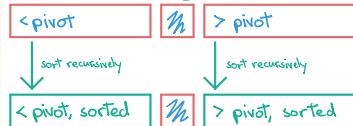
$\Rightarrow$ either $E[Z | x_1 = \text{true}]$ or $E[Z | x_1 = \text{false}]$ is $\geq E[Z]$!

QuickSort,

Random divide and conquer

① pick random 'pivot'

② Divide A into two parts



Expected running time of QuickSort

For each $i, j \in [n]$ w/ $i < j$,

$X_{ij} = \begin{cases} 1 & \text{if } i\text{th smallest element is compared to } j\text{th smallest} \\ 0 & \text{otherwise} \end{cases}$

$\sum_{i < j} X_{ij}$ = total # comparisons made by the algorithm
 $\approx$ total running time

$E[\sum_{i < j} X_{ij}]$ = average running time

$E[\sum_{i < j} X_{ij}] = \sum_{i < j} E[X_{ij}]$ (linearity of expectation)

now analyze $E[X_{ij}]$ in isolation

$E[X_{ij}] = \Pr[X_{ij} = 1]$
 $= \Pr[i\text{th element is compared to } j\text{th}]$

i th and j th elem selected before any element in between happens w/ probability $\frac{2}{j-i+1}$

$E[X_{ij}] = \frac{2}{j-i+1}$

$E[\sum_{i < j} X_{ij}] = \sum_{i < j} \frac{2}{j-i+1}$ (by linearity of expectation)

$\leq 2n \sum_{k=1}^n \frac{1}{k} \approx 2n \ln n$

$\leq O(n \log n)$ with harmonic series

Quickselect (HW)

Lecture 2

Probability

Hash Functions,

Universal Hash Functions

$h: [n] \rightarrow k$ s.t. for all pairs $i, j \in [n]$,
 $\Pr[h(i) = h(j)] \leq \frac{1}{k}$ (much weaker than ideal hash)

e.g. $h(x) = (ax + b) \bmod p \bmod k$

where • p prime, $> k$

• a in $\{1, \dots, p-1\}$ unif. random

• b in $\{0, \dots, p-1\}$ unif. random



Markov's Inequality

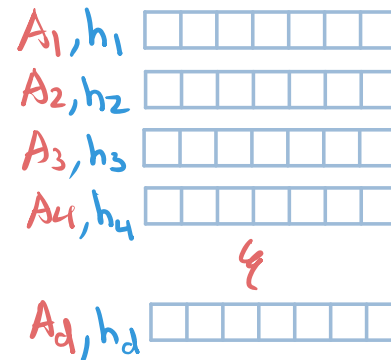
Markov's Inequality (special case)

Let $X \geq 0$ be nonnegative RV

$$\Pr[X \geq 2E[X]] \leq \frac{1}{2}$$

Algorithms

Frequency Estimation, Heavy Hitters (in streams)



estimate f_i w/
 $\min_{j=1, \dots, d} A_j[h_j(i)]$

Lecture 3

Probability

k-wise independence,
higher order moments
(generalizing universal hash
functions and Markov's inequality)

Algorithms

Hash tables w/
chaining +
linear probing

k-wise independent hash functions

$$h: [U] \rightarrow [m]$$

s.t. for any k keys $x_1, x_2, \dots, x_k \in [U]$,
 k values $y_1, y_2, \dots, y_k \in [m]$,

$$P[h(x_1)=y_1, h(x_2)=y_2, \dots, h(x_k)=y_k] = \frac{1}{m^k}$$

eg. $h(x) = a_0 + a_1x + a_2x^2 + \dots + a_{k-1}x^{k-1} \pmod{p}$

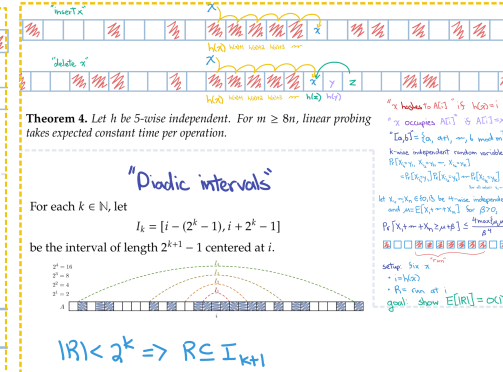
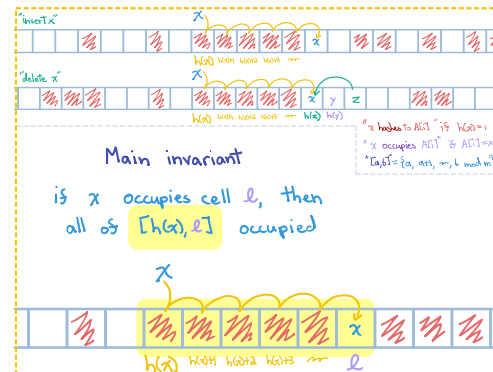
where $p \geq U$ prime,
 $a_0, a_1, \dots, a_{k-1} \in [p]$ indep, uniformly random

Important lemma

k-wise independent random variables
 $P[X_1=y_1, X_2=y_2, \dots, X_k=y_k]$
 $= P[X_1=y_1]P[X_2=y_2] \dots P[X_k=y_k]$
for all values $y_1, \dots, y_k$

let $X_1, \dots, X_n \in \{0, 1\}$ be 4-wise independent
and $\mu = E[X_1 + \dots + X_n]$. then for $\beta > 0$,

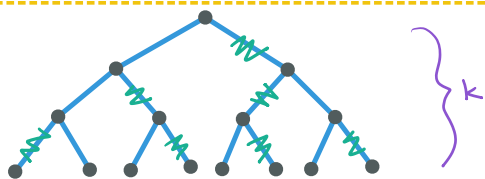
$$P[X_1 + \dots + X_n \geq \mu + \beta] \leq \frac{4 \max\{\mu, \mu^3\}}{\beta^4}$$



Lecture 4

Probability

Galton-Watson processes



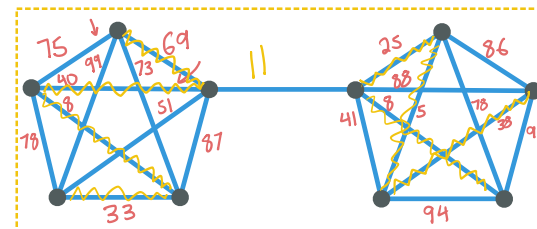
let T = complete binary tree of height k ,
w/ every edge deleted w/ prob $\leq 1/2$

Fact: $P[\text{some leaf connected to root}] \geq \frac{1}{k}$

"Galton-Watson process"

Algorithms

Random Contractions for min cut



Randomized contractions

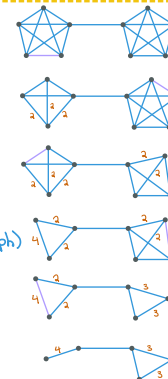
Input: $G=(V,E)$, weights $c \in \mathbb{R}_{>0}^E$
(unweighted)

① while $|E| > 1$

 Ⓐ sample $e \in E$ in proportion to c

 Ⓑ "contract" e

② return set of edges (in original graph)
 contracted to remaining edge



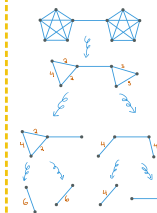
Amplification by branching

Instead of restarting at the beginning, restart partway through

High-level algorithm

① randomly contract until prob.
of ruining the min-cut is $\sim 1/2$

② Branch:
make two copies of current graph
recurse on both copies



Lecture 5

Probability

Multiplicative Chernoff Bound

Multiplicative Chernoff bound

let $X_1, \dots, X_n \in [0, 1]$ be independent

$$\mu = E[X_1 + \dots + X_n]$$

$$\epsilon \in (0, 1)$$

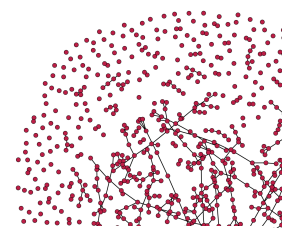
- $P[X_1 + \dots + X_n \geq (1 + \epsilon)\mu] \leq e^{-\epsilon^2 \mu / 3}$
- $P[X_1 + \dots + X_n \leq (1 - \epsilon)\mu] \leq e^{-\epsilon^2 \mu / 2}$

Algorithms (/ results)

Random graphs & the giant component

Erdős-Rényi graphs $G(n, p)$

- n vertices
- each edge sampled independently w/ probability p



Theorem 5.3. Consider a random graph $G \sim \mathcal{G}(n, p)$ for $p = c/n$, where c is a constant.

- If $c > 1$, then with high probability, there is exactly one connected component of G with $\Omega(n)$ vertices, and all other components have size $\leq O(\log n)$.
- For $c < 1$, then with high probability, all connected components of G has size $< O(\log n)$.

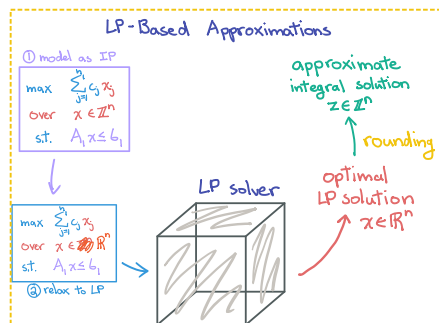
Lecture 6

Probability (& techniques)

Approximations via linear programming & randomized rounding

Linear programs (LPs)
Similar to IPs, except no integrality constraints

min/max $\sum_{j=1}^n c_j x_j$ linear objective
over $x \in \mathbb{R}^n$ linear integral variables
st. $A_1 x \leq b_1$ linear constraints
 $A_2 x = b_2$
 $A_3 x \geq b_3$



Continuous optimization vs Discrete optimization

randomization

Algorithms

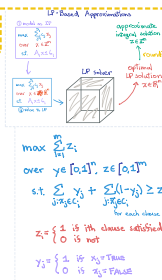
Max Sat Set Cover Covering Integer Programs

For each variable x_j , independently, randomly set $x_j = 1$ with probability p_j , and $x_j = 0$ otherwise.

$$E[\# \text{ sat. clauses}] \geq (1 - 1/e) \text{OPT}$$

k	oblivious	LP	average
1	1/2	$z_i \geq (3/4)z_i$	$\geq (3/4)z_i$
2	3/4	$(3/4)z_i \geq (3/4)z_i$	$\geq (3/4)z_i$
3	7/8	$(1 - 1/e)z_i \geq (3/4)z_i$	$\geq (3/4)z_i$
4	15/16	$(1 - 1/e)z_i \geq (3/4)z_i$	$\geq (3/4)z_i$
5	31/32	$(1 - 1/e)z_i \geq (3/4)z_i$	$\geq (3/4)z_i$
...	...	...	...
k	$1 - 1/2^k$	$(1 - 1/e)z_i \geq (3/4)z_i$	$\geq (3/4)z_i$

Can we get $\frac{3}{4}$ -APX?



Rounding LP solution
 $x \in [0, 1]^n \rightsquigarrow F \subseteq \{S_1, \dots, S_n\}$

For each set S_j , independently, take S_j with probability x_j .

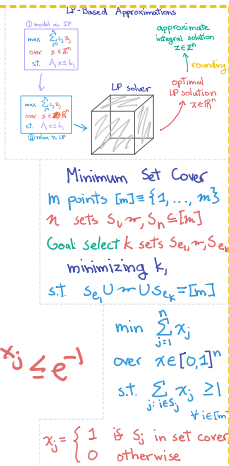
$$E[|F|] = \sum x_j \leq \text{OPT}$$

Fix a point i .

$$E[\# \text{ sets in } F \text{ covering } i] = \sum_{j \in S_i} x_j \geq z_i$$

$$P[F \text{ not cover } i] = \prod_{j \in S_i} (1 - x_j) \leq e^{-\sum_{j \in S_i} x_j} \leq e^{-z_i}$$

$$P[F \text{ cover } i] \geq 1 - e^{-1}$$



Lecture 7

Probability

Continuous Random Variables

Variance, Chebyshev's Variance Reduction

Median trick

Variance

$$\text{Var}(X) = E[(X - E[X])^2] = E[X^2] - (E[X])^2$$

Chebyshev's inequality

$$P[|X - E[X]| \geq d] \leq \frac{\text{Var}(X)}{d^2}$$

"median trick", "median of means"

- ① take mean to reduce variance, get constant prob. of error
- ② take median to amplify success prob. exponentially

Algorithms

Distinct Elements



Goal: estimate # of distinct elements in the stream

Suppose you had an ideal hash function

$$h: [m] \rightarrow [0, 1]$$

and the following facts:

let $Y_1, \dots, Y_k \in [0, 1]$ independent and uniformly random

$$X = \min \{Y_1, \dots, Y_k\}$$

$$\text{then } E[X] = \frac{1}{k+1}$$

$$\textcircled{1} X \leftarrow 1$$

② for each element e

$$\textcircled{a} X \leftarrow \min \{X, h(e)\}$$

$$/* E[X] = \frac{1}{k+1}, k = (\text{\# distinct elements}) */$$

$$\textcircled{3} \text{ return } \frac{1}{X} - 1$$

$$\text{recall } E[X] = \frac{1}{k+1}$$

$$\text{goal: show } |X - E[X]| \leq \frac{\epsilon}{k+1}$$

w/ good prob.

$$P[|X - E[X]| \geq \frac{\epsilon}{k+1}]$$

$$\leq \text{Var}(X) \cdot \left(\frac{k+1}{\epsilon}\right)^2 \leftarrow \text{Chebyshev's inequality}$$

where

$$\text{Var}(X) = \frac{k}{(k+1)^2(k+2)} \approx \frac{1}{k^2}$$

Q: how to reduce $\text{Var}(X)$?

we can get $(1 \pm \epsilon)$ -APX

w/ 10% error prob. in $O(\frac{1}{\epsilon^2})$ space

what about high probability?

① make q copies $Y_1, \dots, Y_q$ of

② return median of $Y_1, \dots, Y_q$

Algo correct $\Leftrightarrow \geq \frac{1}{2}$ of Y_i 's correct

let $Z = \# \text{correct } Y_i$'s $E[Z] = \frac{q}{2}$

$$P[Z \leq (1 - \frac{\epsilon}{4})E[Z]] \leq e^{-\left(\frac{\epsilon}{4}\right)^2 E[Z]/2} \leq \frac{1}{n^{O(n)}} \text{ for } q = O(\log n)$$

ideal hash $h: [m] \rightarrow [0, 1]$

let $Y_1, \dots, Y_k \in [0, 1]$

indep. and unifs. rand.

$$X = \min \{Y_1, \dots, Y_k\}$$

$$\text{then } E[X] = \frac{1}{k+1}$$

$$\text{and } E[X^2] = \frac{2}{(k+1)(k+2)}$$

$$\text{Var}(X) = E[(X - E[X])^2]$$

$$X_1, \dots, X_q \text{ indep. copies of } X$$

$$\bar{X} = \frac{1}{q}(X_1 + \dots + X_q)$$

$$\text{Var}(\bar{X}) = \frac{1}{q} \text{Var}(X)$$

$$P[X_1 + \dots + X_q \leq (1 - \epsilon)q] \leq e^{-\epsilon^2 q/4}$$

Lecture 8

Probability

Gaussian random variables

Gaussian random variables

$$x \sim \mathcal{N}(\mu, \sigma^2)$$

mean variance

Def: the nicest, most pleasant, convenient random variable ever

Fact let $x \sim \mathcal{N}(\mu, \sigma^2)$, $d \in \mathbb{R}$.

$$\text{then } x+d \sim \mathcal{N}(\mu+d, \sigma^2)$$

$$dx \sim \mathcal{N}(d\mu, d^2\sigma^2)$$

Fact let $x_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$, $x_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$

$$\text{then } x_1 + x_2 \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$$

Fact let $x \sim \mathcal{N}(0, 1)$, $t < \frac{1}{2}$

$$\text{then } E[e^{tx^2}] = \frac{1}{\sqrt{1-2t}}$$

Algorithms

Dimensionality reduction

Bag-of-words

literally any sentence is fine

any ↑ is literally sentence English
five encodes text in high dimensions

Lemma 8.8. Let $A \sim \mathcal{N}^{k \times d}$ be a random matrix where each coordinate A_{ij} is an independently drawn sample from $\mathcal{N}$. Let $\epsilon > 0$ be sufficiently small. For any vector x ,

$$\mathbb{P}\left[(1-\epsilon)\|x\|^2 \leq \frac{1}{k}\|Ax\|^2 \leq (1+\epsilon)\|x\|^2\right] \geq 1 - 2e^{-k/8}$$

$$k \left\{ \left[\frac{1}{k} \|Ax\|^2 \right] \leftarrow \alpha_i \right.$$

$$E[\|Ax\|^2] = \sum_{i=1}^k E[(Ax)_i^2] = k\|x\|^2$$

$$E[(\sum_{i=1}^k A_{ij} x_j)^2] = \|x\|^2$$

$$A_{ij} x_j \sim \mathcal{N}(0, x_j^2)$$

$$\sum_{j=1}^n A_{ij} x_j \sim \mathcal{N}(0, \|x\|^2)$$

$$(Ax) \sim \mathcal{N}^k(0, \|x\|^2)$$

$$\Pr[\|Ax\|^2 \geq k\|x\|^2 \cdot (1+\epsilon)] \leq e^{-\epsilon^2 k/8}$$

Dimensionality Reduction - high level

$O(\log n)$ dimensions suffice

where $n = \# \text{ points}$

$$\mathbb{R}^{\text{very large}} \xrightarrow{f(x)} \mathbb{R}^{O(\log n)}$$

One can map n points into $\mathbb{R}^{O(\log n/\epsilon^2)}$ while preserving all distances up to $(1 \pm \epsilon)$ -APX

Lecture 9

Probability

Locality-sensitive hashing

Fix $\theta \in [0, \pi]$. given q , either:

(a) output $p \in P$ w/ $\angle(p, q) \leq 4\theta$

(b) declare no $p \in P$ w/ $\angle(p, q) \leq \theta$

define $h: S^{d-1} \rightarrow \{-1, 1\}^k$ by

$$h_i(x) = \text{sign}(\langle g_i, x \rangle) \text{ w/ } g_i \sim N^d$$

$$k = \pi(\log(n)/2\theta)$$



d-dimensional sphere

when querying y :

- if there is $x \in P$ w/ $\angle(x, y) \leq \theta$, then it collides w/ probability $\geq 1/\sqrt{n}$

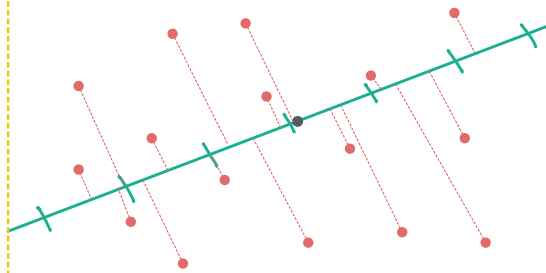
- expect to collide w/ at most one point $x \in P$ w/ $\angle(x, y) > 2\theta$

Lemma let $x, y \in S^{d-1}$
then $g \sim N^d$

$$P[\text{sign}(\langle g, x \rangle) \neq \text{sign}(\langle g, y \rangle)] = \frac{\angle(x, y)}{\pi}$$

Random line embedding & Buckets

$$h(x) = \lfloor \langle g, x \rangle + d \rfloor \text{ w/ } g \sim N^d, d \in [0, 1] \text{ unifs. rand.}$$

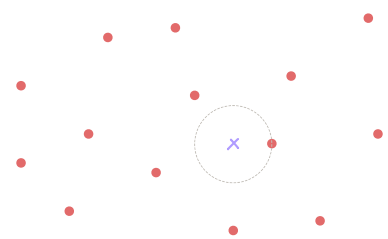


Algorithms

Approximate nearest neighbors

Nearest neighbor search

Input: n points $P \subset \mathbb{R}^d$



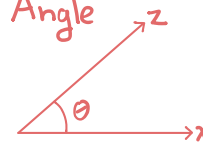
Query: given $q \in \mathbb{R}^d$, find closest point in P to q

Types of distances

- Euclidean distance

$$\|x - z\|_2 = \sqrt{\sum_i (x_i - z_i)^2}$$

- Angle



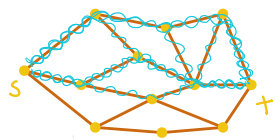
Lecture 10

Probability (and techniques)

LP duality

L_1 -metric embeddings

Max Flow Min Cut



Path Packing LP

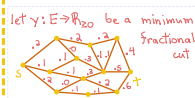
maximize $\sum_{p \in P} x_p$
over $x: P_{st} \rightarrow \mathbb{R}_{\geq 0}$
(all s-t paths)

s.t. $\sum_{p \in P} x_p \leq c_e$ for all edges e
"Ax ≤ c" (capacity of e)

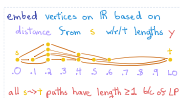
Path Covering LP (i.e. cut)

minimize $\sum_{e \in E} c_e y_e$
over $y: E \rightarrow \mathbb{R}_{\geq 0}$
(total capacity of "fractional cut")

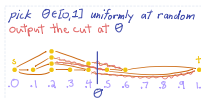
s.t. $\sum_{e \in P} y_e \geq 1$ for all $p \in P_{st}$
"A'y ≥ b" (continuous analog of s-t cut)



let $y: E \rightarrow \mathbb{R}_{\geq 0}$ be a minimum fractional cut



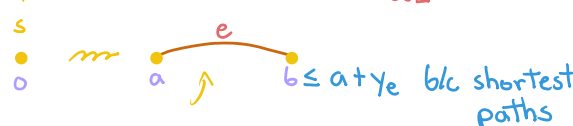
embed vertices on $\mathbb{R}$ based on distance from s with lengths y



pick $\theta \in [0, 1]$ uniformly at random output the cut at θ

$E[\text{capacity of cut}]$

$$\stackrel{\text{(linearity of expectation)}}{=} \sum_{e \in E} c_e \Pr[e \text{ cut}] \leq \sum_{e \in E} c_e y_e$$



minimum capacity of any fractional cut

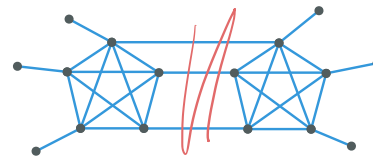
b/c shortest paths

Algorithms

Sparsest Cut

Special case: uniform sparsest cut
w/ uniform demands $b(u,v) = 1 \quad \forall u,v$

$$\text{"sparsity"} = \frac{c(\delta(s))}{|S| |T|}$$



useful primitive for graph partitioning, divide & conquer

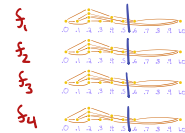
Q: how to round LP to sparse cut?

min $\sum_{e \in E} c_e d(u,v)$ over metrics $d: V \times V \rightarrow \mathbb{R}_{\geq 0}$
s.t. $\sum_{i=1}^k b_i d(s_i, t_i) \geq 1$

Suppose $d(u,v)$ was of the form

$$d(u,v) = \sum_{i=1}^k |f_i(u) - f_i(v)| \quad \text{for } f_i: V \rightarrow [0, 1]^k$$

(assume $\min f_i(a) = 0, \max f_i(a) = 1$)



" L_1 -metric"

Sparsest cut
Input: undirected graph $G=(V,E)$, edge capacities $c(e) > 0$ for $e \in E$, demands $b(u,v) > 0$ for $u,v \in V$.
Goal: compute SCF minimizing $\frac{c(\delta(s))}{\sum_{i=1}^k b_i d(s_i, t_i)}$ ← cut has many free variables

Lecture 11

Probability (and techniques)

Algorithms

Other metric-based problems
(sometimes via correspondence between cuts $\leftrightarrow$ metrics)

Buy-at-bulk network design
k-median clustering
metric labeling
group Steiner tree
metrical task systems
vehicle routing
hierarchical placement
distributed k-server
covering Steiner tree
⋮



these problems are easy for tree metrics

Approximation by tree metrics

Input: undirected $G=(V,E)$ w/ edge weights $w: E \rightarrow \mathbb{R}_{\geq 0}$

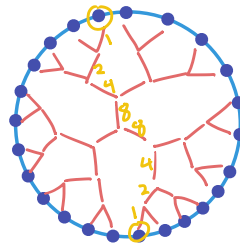
(High-level)
Goal: compute tree T (containing V as nodes)
so that shortest paths on T are close to T

$$d_T(u,v) \geq d_G(u,v)$$

$$\text{For } e=f_u,v \quad \begin{matrix} \text{(distance in } T) \\ \text{(stretch } e) = \frac{d_T(u,v)}{d_G(u,v)} \\ \text{(distance in } G) \end{matrix}$$

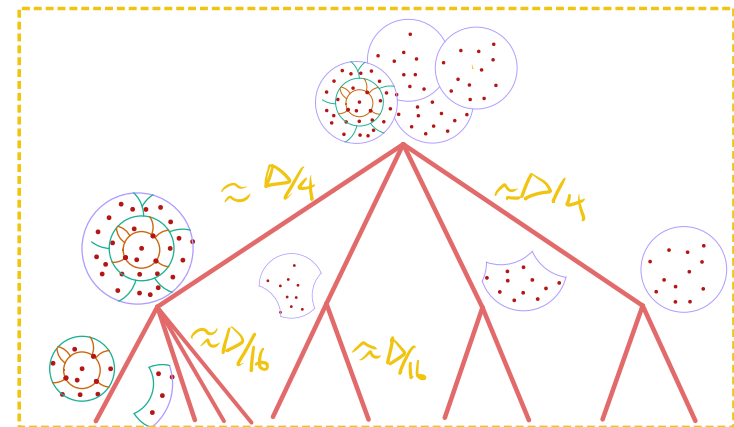
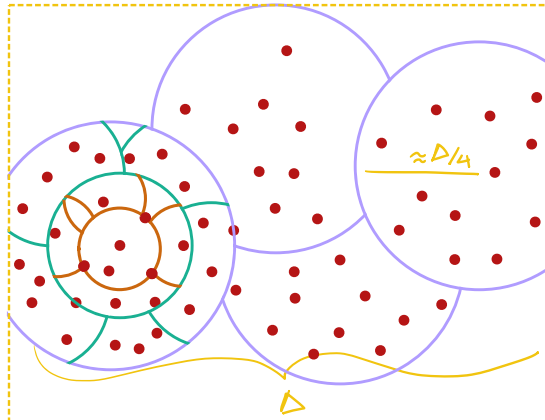
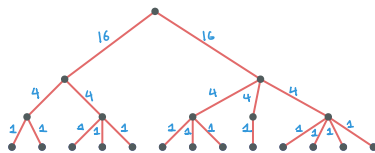
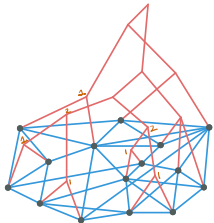
ideally

$$(\text{stretch } e) = \text{polylog}(n) \quad \forall e$$



Randomized Tree metrics

Theorem In randomized polynomial time,
randomized hierarchical dominating tree metric w/
 $E[\text{stretch } e] \leq O(\log n) \quad \forall e \in E$



Lecture 12

Probability (and techniques)

Additive Chernoff Inequality

Double-sampling

VC dimension

Additive Chernoff bound

let $X_1, \dots, X_n \in [0, 1]$ be independent
 $\mu = E[X_1 + \dots + X_n] / n$
 $\epsilon \in (0, 1)$

- $P[\frac{X_1 + \dots + X_n}{n} \geq \mu + \epsilon] \leq e^{-2\epsilon^2 n}$
- $P[\frac{X_1 + \dots + X_n}{n} \leq \mu - \epsilon] \leq e^{-2\epsilon^2 n}$

Algorithms

ϵ -samples and ϵ -nets

ϵ -sample $|M_Q(r) - \mu(r)| \leq \epsilon \quad \forall r \in R$

let Q sample k points w/ repetition
 $k = O\left(\frac{\ln(g(|P|))}{\epsilon^2}\right) \Rightarrow \epsilon$ -sample $k = O\left(\frac{\ln(g(|P|))}{\epsilon^2}\right) \Rightarrow \epsilon$ -net (w/ constant probability)

ϵ -sample theorem [VC71]
 $k \geq O\left(\frac{\ln(g(2k))}{\epsilon^2}\right) \Rightarrow \epsilon$ -sample (w/ constant probability)

e.g. $g(n) = O(n^d)$
 $k \geq O\left(\frac{d \ln(d/\epsilon)}{\epsilon^2}\right) \Rightarrow \epsilon$ -sample
 ↑ independent of $|P|$ (!)

$P[\frac{X_1 + \dots + X_n}{n} \geq \mu + \epsilon], P[\frac{X_1 + \dots + X_n}{n} \leq \mu - \epsilon] \leq e^{-2\epsilon^2 n}$ $g(n) = \max\{|RAQ| : a \in P, |a| \leq n\}$

Diagram: Points P and ranges R (circles). A range r is shown with its measure $\mu(r) = \frac{|P \cap r|}{|P|}$ and its sample measure $M_Q(r) = \frac{|a \cap r|}{|a|}$. A red 'X' indicates a bad range where the difference is large.

ϵ -sample $|M_Q(r) - \mu(r)| \leq \epsilon \quad \forall r \in R$

let Q sample k points w/ repetition
 $k = O\left(\frac{\ln(g(|P|))}{\epsilon^2}\right) \Rightarrow \epsilon$ -sample $k = O\left(\frac{\ln(g(|P|))}{\epsilon^2}\right) \Rightarrow \epsilon$ -net (w/ constant probability)

ϵ -net theorem [HW87]
 $k \geq O\left(\frac{\ln(g(2k))}{\epsilon^2}\right) \Rightarrow \epsilon$ -net (w/ constant probability)

e.g. $g(n) = O(n^d)$
 $k \geq O\left(\frac{d \ln(d/\epsilon)}{\epsilon^2}\right) \Rightarrow \epsilon$ -net
 ↑ independent of $|P|$ (!)

$P[\frac{X_1 + \dots + X_n}{n} \geq \mu + \epsilon], P[\frac{X_1 + \dots + X_n}{n} \leq \mu - \epsilon] \leq e^{-2\epsilon^2 n}$ $g(n) = \max\{|RAQ| : a \in P, |a| \leq n\}$

Diagram: Points P and ranges R (circles). A range r is shown with its measure $\mu(r) = \frac{|P \cap r|}{|P|}$ and its sample measure $M_Q(r) = \frac{|a \cap r|}{|a|}$. A red 'X' indicates a bad range where the difference is large.

Lecture 13

Probability
(and techniques)

Algorithms
(models and applications)

Consistency model

PAC learning model

Generalization error

Occam's Razor

Generalization by growth

Lecture 14

Probability
(and techniques)

Algorithms
(models and applications)

Entropy

(Optimal)
Error-Correcting
Codes

Lecture 15

Probability (and techniques)

Algorithms (models and applications)

Lovász Local Lemma

Moser's resampling
algorithm

Lecture 16

Probability (and techniques)

Algorithms (models and applications)

Random Walks

Stationary Distribution

Perron-Frobenius Theorem

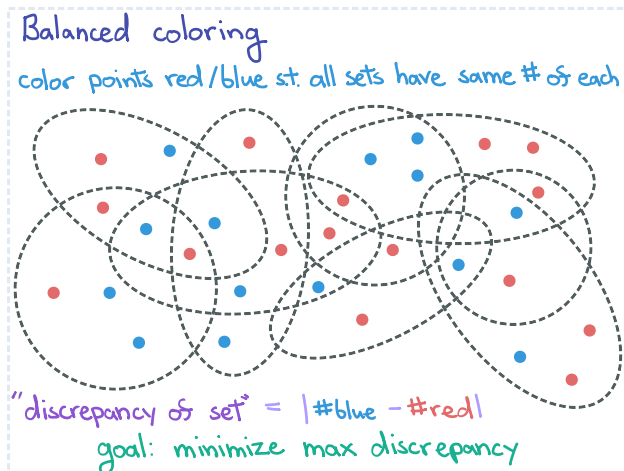
Pagerank

Lecture 17

Probability (and techniques)

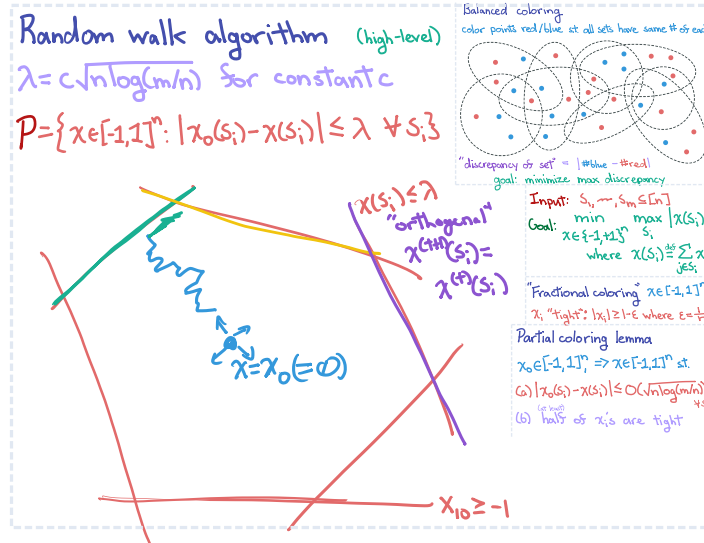
High-dimensional random walks

More fun w/ Gaussians



Algorithms (models and applications)

Discrepancy minimization



Lecture 18

Probability (and techniques)

Algorithms (models and applications)

Hitting times

Cover times

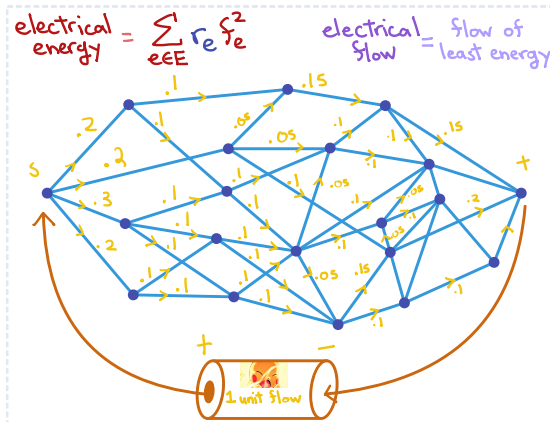
Electrical networks

Connectivity
in $O(\log n)$ space

random walk terminology:

Hitting time $H(s, t)$ = expected # steps until a random walk from s reaches t

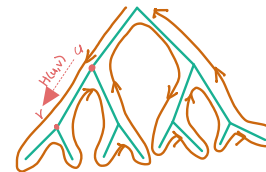
Cover time $C(s)$ = expected # steps until a random walk from s visits every vertex



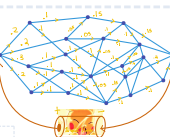
Lemma $H(u, v) + H(v, u) = 2m \cdot (\text{energy of unit flow from } u \text{ to } v)$

Theorem $C(s) \leq 2mn$

proof Six traversal of spanning tree



$$\begin{aligned} \text{total time} &= \sum_{u, v \in V} H(u, v) + H(v, u) \\ &= 2m \sum_{u, v \in V} (\text{energy of unit flow from } u \text{ to } v) \leq 2m(n-1) \end{aligned}$$



electrical energy = $\sum_{e \in E} r_e f_e^2$
electrical flow = least energy

des $B: \mathbb{R}^E \rightarrow \mathbb{R}^V$ by
(BS) $_v$ = net flow at v
 $d_v = \begin{cases} -1 & \text{if } v = s \\ 1 & \text{if } v = t \\ 0 & \text{otherwise} \end{cases}$

min $\frac{1}{2} \sum_{e \in E} r_e f_e^2$ s.t. $BS = d$
(electrical energy) (1 is optimal)

effective resistance of d = optimum value/energy
Ohm's law: $f = R^{-1} B^T p$
for electric flow by electric potentials p
 $L = BR^{-1}B^T \in \mathbb{R}^{V \times V}$
(effective resist. of d) $= \langle d, L^{-1}d \rangle$

Lecture 19

Probability (and techniques)

Algorithms (models and applications)

Spectral theorem
for symmetric maps

Spectral analysis of
random walks

Lecture 20

Probability
(and techniques)

Algorithms
(models and applications)

Conductance

Cheeger's inequality

Conductance-based
mixing time

Fiedler's algorithm
for low-conductance
cuts

Lecture 21

Probability
(and techniques)

Algorithms
(models and applications)

Expanders

Zigzag product

tensors

Deterministic

Connectivity in

$O(\log n)$ space

Expander graphs

G is an expander if the spectral gap γ is at least a constant

Theorem, restated: let G be a dir. graph and R the random walk linear map. Then R has an eigenvector $x \in \Delta^V$ w/ eigenvalue 1. If G is strongly connected, then x is the unique eigenvector w/ eigenvalue 1

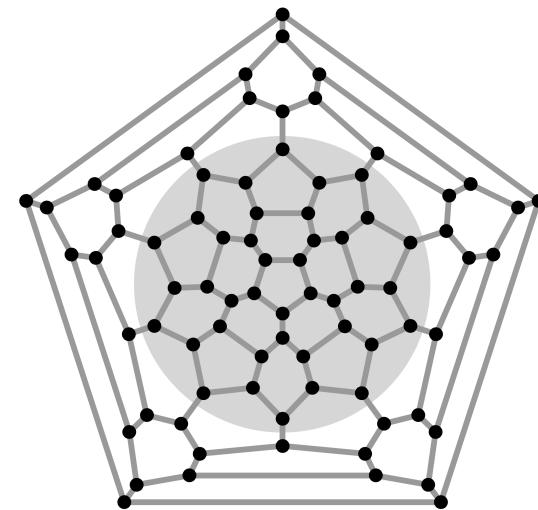
Theorem: $G=(V,E)$ undirected, $w:E \rightarrow \mathbb{R}_{>0}$
let $R: \mathbb{R}^V \rightarrow \mathbb{R}^V$ be random walk map
then R has stationary distr. $s \in \Delta_V$ w/
 $s_v = \frac{\deg(v)}{2W}$ $\deg(v) = \sum_{e \in \mathcal{E}(v)} w(e)$
 $W = \sum_{e \in E} w(e)$

Theorem. let $\gamma = 1 - \max\{\mu_2, |\mu_n|\}$
 $\|R^k x - s\| \leq (1-\gamma)^k \sqrt{\frac{\Delta_{\max}}{\Delta_{\min}}}$
stationary distribution
 $1 \geq \mu_2 \geq \dots \geq \mu_n = \text{eigenvalues of } R$

Q: deterministic connectivity for constant-degree expanders?

- Diameter $O(\log n)$
- $O(1)$ random bits per random step

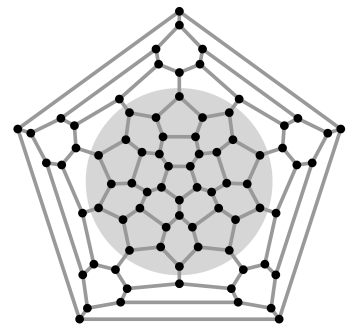
$\Rightarrow$ enumerate all paths of length $O(\log n)$!



Theorem: $O(\log n)$ space, poly-time [Reingold, 2005]
deterministic connectivity in undirected graphs.

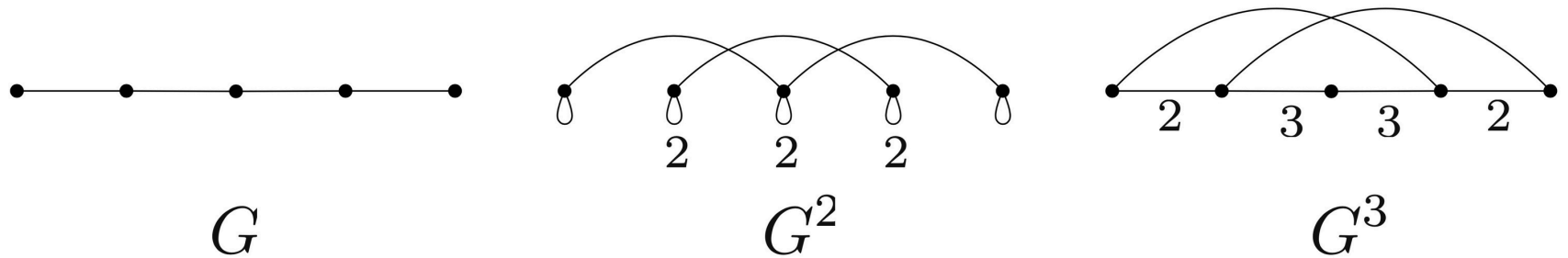
High level idea:

- convert G into $O(1)$ -degree expander
by sequence of "graph transformations"
(graph transformations applied implicitly due to space constraints)
- derandomize the random walk
algo (implicitly) on the expander



Lemma: $O(\log n)$ space, poly-time
deterministic connectivity in
 $O(1)$ -degree expanders

Powering



$G^k =$ graph w/ an edge for every k step walk in G
(w/ repetition)

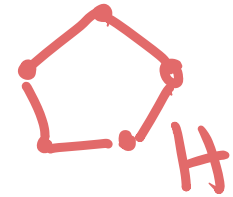
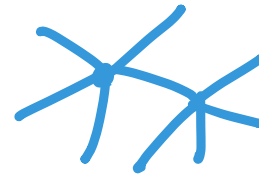
random walk R^k has eigenvalues $1, \mu_2^k, \dots, \mu_n^k$

$$\delta(G^k) = 1 - (1 - \delta(G))^k \quad \text{😊}$$

degree increases from d to d^k 😞

Zig-zag product $Z(G, H)$

let G be d -regular

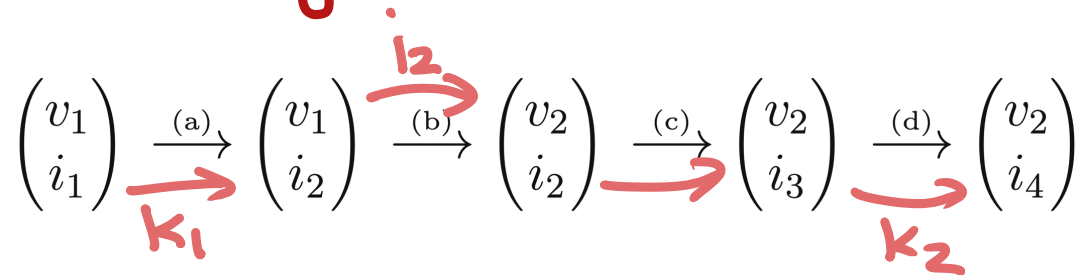


H be d_0 -regular w/ d_0 vertices (H will be $O(D)$ -degree expander)

vertices: $V_G \times V_H = \{(u, v) : u \in V_G, v \in V_H\}$

edges: for $(v_1, i_1) \in V_G \times V_H$, $k_1, k_2 \in [d_0]$:

the (k_1, k_2) th neighbor of (v_1, i_1) is (v_2, i_4) where:



where:

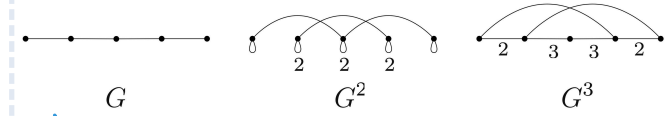
- (a) is a step in H from i_1 to its k_1 th neighbor, i_2 .
- (b) moves v_1 to the i_2 th outgoing neighbor of v_1 , v_2 .
- (c) moves from i_2 to i_3 where v_1 is the i_3 th neighbor of v_2 .
- (d) moves i_3 to its k_2 th neighbor in H .

(so $Z(G, H)$ is d_0^2 -regular)

	G^k	$Z(G, H)$
vertices	n	nd
degree	d^k	d_0^2
spectral gap	$1 - (1 - \gamma_G)^k$	$\gamma_G \gamma_H^2$

	$Z(G^2, H)$
vertices	nd
degree	d_0^2
spectral gap	$(1 - (1 - \gamma_G^2)) \gamma_H^2 = (2\gamma_G - \gamma_G^2) \cdot \gamma_H^2$

Powering



G^k = graph w/ edge for every k step walk in G
(w/ repetition)

random walk R^k has eigenvalues $1, \mu_2^k, \dots, \mu_n^k$

$$\gamma(G^k) = 1 - (1 - \gamma(G))^k \quad \text{😊}$$

degree increases from d to d^k 😞

Zig-zag product $Z(G, H)$

let G be d -regular

H be d_0 -regular w/ d vertices (H will be $O(d)$ -degree expander)

vertices: $V_G \times V_H = \{(u, v) : u \in V_G, v \in V_H\}$

edges: for $(v_1, i_1) \in V_G \times V_H$, $k_1, k_2 \in [d_0]$:

the (k_1, k_2) th neighbor of (v_1, i_1) is (v_2, i_2) where:

$$\begin{pmatrix} v_1 \\ i_1 \end{pmatrix} \xrightarrow{(a)} \begin{pmatrix} v_1 \\ i_2 \end{pmatrix} \xrightarrow{(b)} \begin{pmatrix} v_2 \\ i_2 \end{pmatrix} \xrightarrow{(c)} \begin{pmatrix} v_2 \\ i_3 \end{pmatrix} \xrightarrow{(d)} \begin{pmatrix} v_2 \\ i_4 \end{pmatrix}$$

where:

- (a) is a step in H from i_1 to its k_1 th neighbor, i_2 .
- (b) moves v_1 to the i_2 th outgoing neighbor of v_1 , v_2 .
- (c) moves from i_2 to i_3 where v_1 is the i_3 th neighbor of v_2 .
- (d) moves i_3 to its k_2 th neighbor in H .

(so $Z(G, H)$ is d_0^2 -regular)

Lecture 22

Probability
(and techniques)

classes RP and BPP

tensors of graphs

Algorithms
(models and applications)

More efficient
amplification

Efficient construction
of exponentially
large expanders

A language L is in...

P if there is a (det.) polytime algo to decide if $x \in L$
(polynomial time)

RP if there's a random. polytime algo that, on input x :
(randomized polynomial time)

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

BPP if there's a random. polytime algo that, on input x :
(bounded error probabilistic polynomial time)

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

A language L is in...

no error

P if there is a (det.) polytime algo to decide if $x \in L$
(polynomial time)

one-sided
error

RP if there's a random. polytime algo that, on input x :
(randomized polynomial time)

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

two-sided
error

BPP if there's a random. polytime algo that, on input x :
(bounded error probabilistic polynomial time)

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

no error

one-sided
error

two-sided
error

A language L is in...

P if there is a (det.) polytime algo to decide if $x \in L$
(polynomial time)

RP if there's a random. polytime algo that, on input x :
(randomized polynomial time)

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

BPP if there's a random. polytime algo that, on input x :
(bounded error probabilistic polynomial time)

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

$$P \subseteq RP \subseteq BPP$$

we don't know if they are distinct!

major question of field of pseudorandomness

Amplification

RP if there's a random. polytime algo that, on input x :
(randomized polynomial time)

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

let LERP

$\Rightarrow$ algo w/ false-negative error prob. (say) $1/2$
using (say) m random bits

how to reduce error prob. to δ ?

1. run algo $O(\log(1/\delta))$ times
2. Output disjunction (or) of all trials

total # of random bits?

$$O(m \log(1/\delta))$$

Better?

Amplification

let **LEBPP**

BPP is there's a random. polytime algo that, on input x :
(bounded error
probabilistic polynomial time)

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

$\Rightarrow$ algo w/ error prob. $1/3$ on both sides
using (say) m random bits

how to reduce error prob. to δ ?

1. run algo $O(\log(1/\delta))$ times
2. Output majority decision over all trials

total # of random bits?

$$O(m \log(1/\delta))$$

Better?

RP if there's a random. polytime algo that, on input x :
(randomized polynomial time)

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

BPP if there's a random. polytime algo that, on input x :
(bounded error probabilistic polynomial time)

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits

ideas?

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits

RP if there's a random. polytime algo that, on input x :

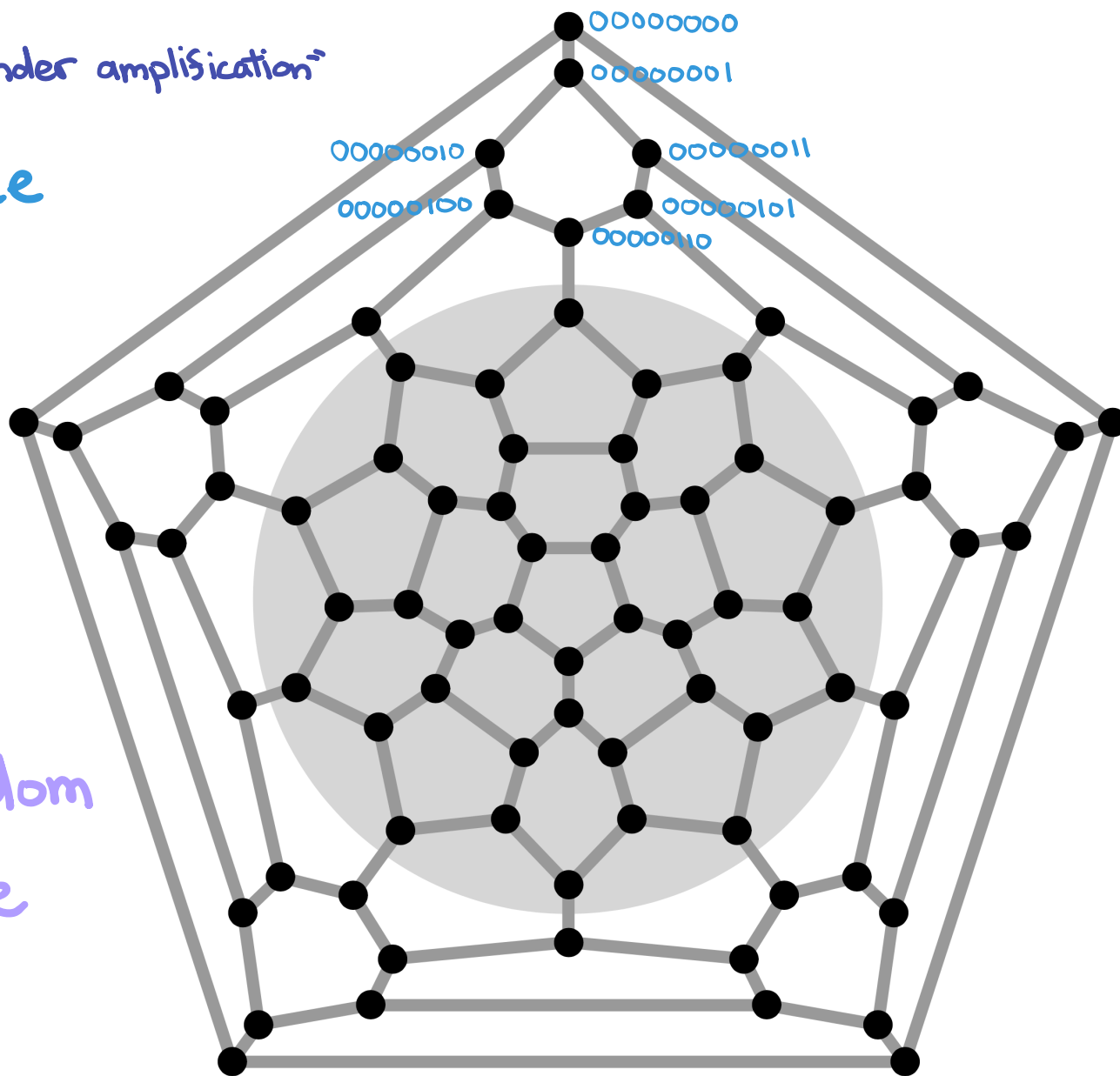
- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

BPP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

High level idea "expander amplification"

- make constant degree expander $G=(V,E)$ w/ $V = \{0,1\}^m$
- take random walk $v_1, \dots, v_t \in \{0,1\}^m$ w/ $t = O(\log(1/\delta))$
- use each v_i as random bits for i th instance of algo



Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

BPP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

constructed implicitly b/c its so big

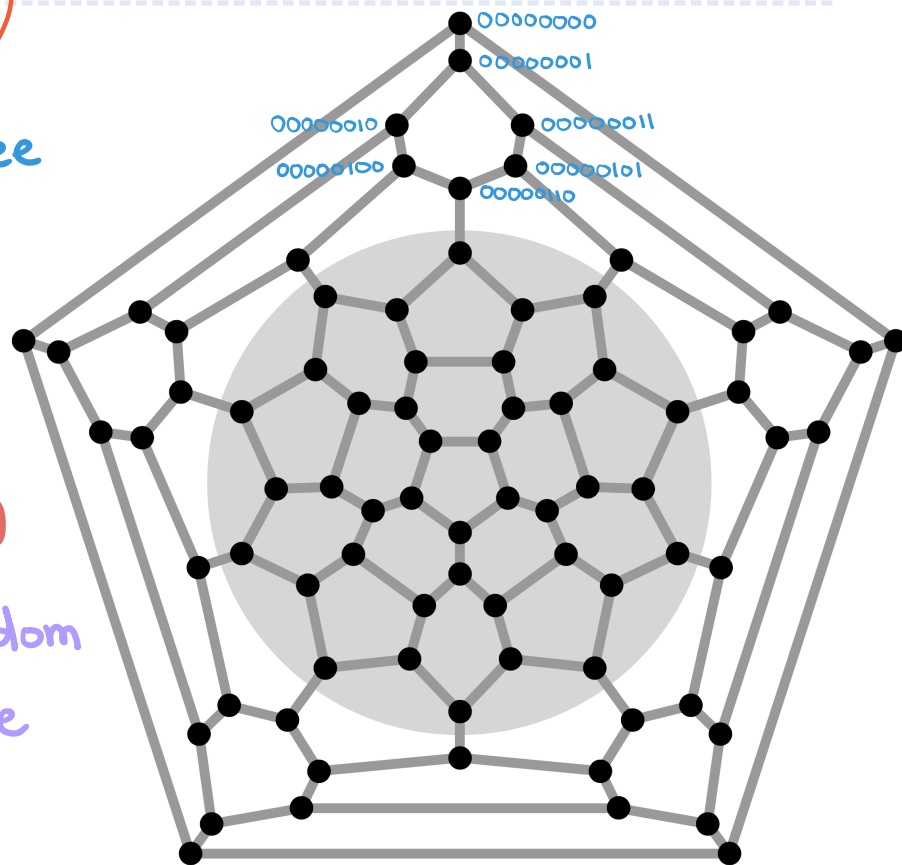
m random bits
to sample v_i

$O(1)$ random bits
per $v_i, i \geq 1$

intuitively,
random walk on
expander is like
uniform sampling

High level idea

- make constant degree expander $G=(V,E)$ w/ $V = \{0,1\}^m$
- take random walk $v_1, \dots, v_t \in \{0,1\}^m$ w/ $t = O(\log(1/\delta))$
- use each v_i as random bits for i th instance of algo



High level idea

- make constant degree expander $G=(V,E)$ w/ $V=\{0,1\}^m$
- take random walk $v_1, \dots, v_t \in \{0,1\}^m$ w/ $t=O(\log(1/\delta))$
- use each v_i as random bits for i th instance of algo

consider RP (one-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

$$\frac{|B|}{|V|} \leq \frac{1}{2} \quad (\text{for example})$$

expander amplification fails iff entire random walk is in B

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

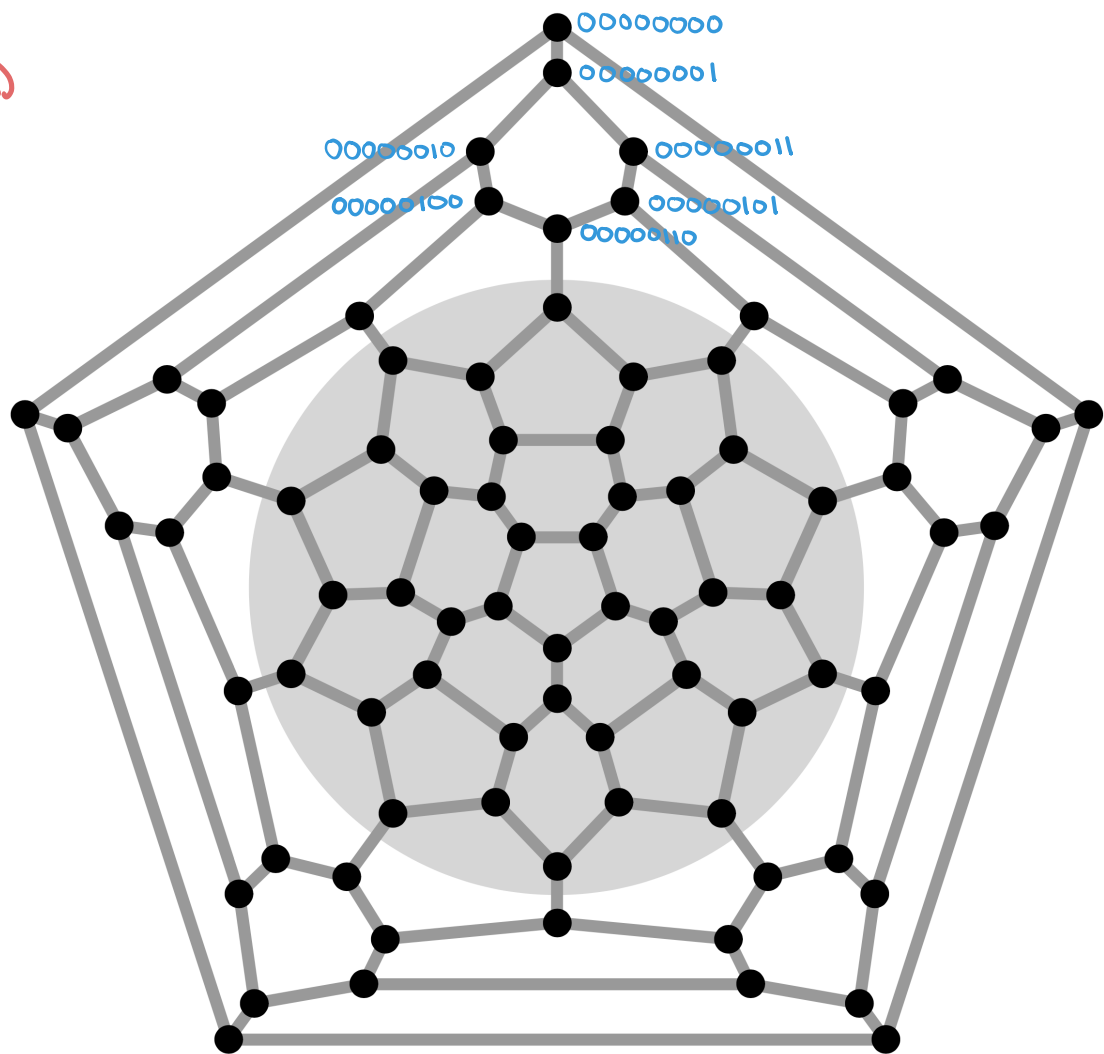
BPP is there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits



High level idea

- make constant degree expander $G=(V,E)$ w/ $V=\{0,1\}^m$
- take random walk $v_1, \dots, v_t \in \{0,1\}^m$ w/ $t = O(\log(1/\delta))$
- use each v_i as random bits for i th instance of algo

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

BPP is there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits

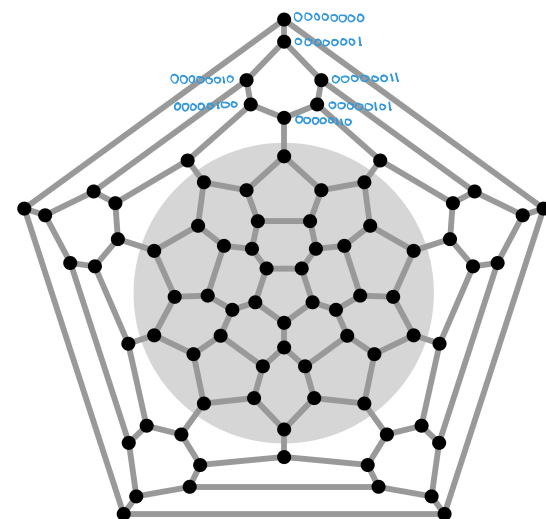
Lemma

G regular, undirected w/ spectral gap γ

$B \subset V$, $\mu = |B|/|V|$

random walk $v_1, \dots, v_t$ (v_1 unif. random)

$$P_r[v_1, \dots, v_t \in B] \leq (1 - (1 - \mu)\gamma)^t$$



consider RP (one-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

expander amplification
fails iff entire random
walk is in B

High level idea

- make constant degree expander $G=(V,E)$ w/ $V=\{0,1\}^m$
- take random walk $v_1, \dots, v_t \in \{0,1\}^m$ w/ $t=O(\log(1/\delta))$
- use each v_i as random bits for i th instance of algo

consider **BPP** (two-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

$$\frac{|B|}{|V|} \leq \frac{1}{3} \quad (\text{for example})$$

expander amplification

fails iff **at least half** of random walk is in B

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

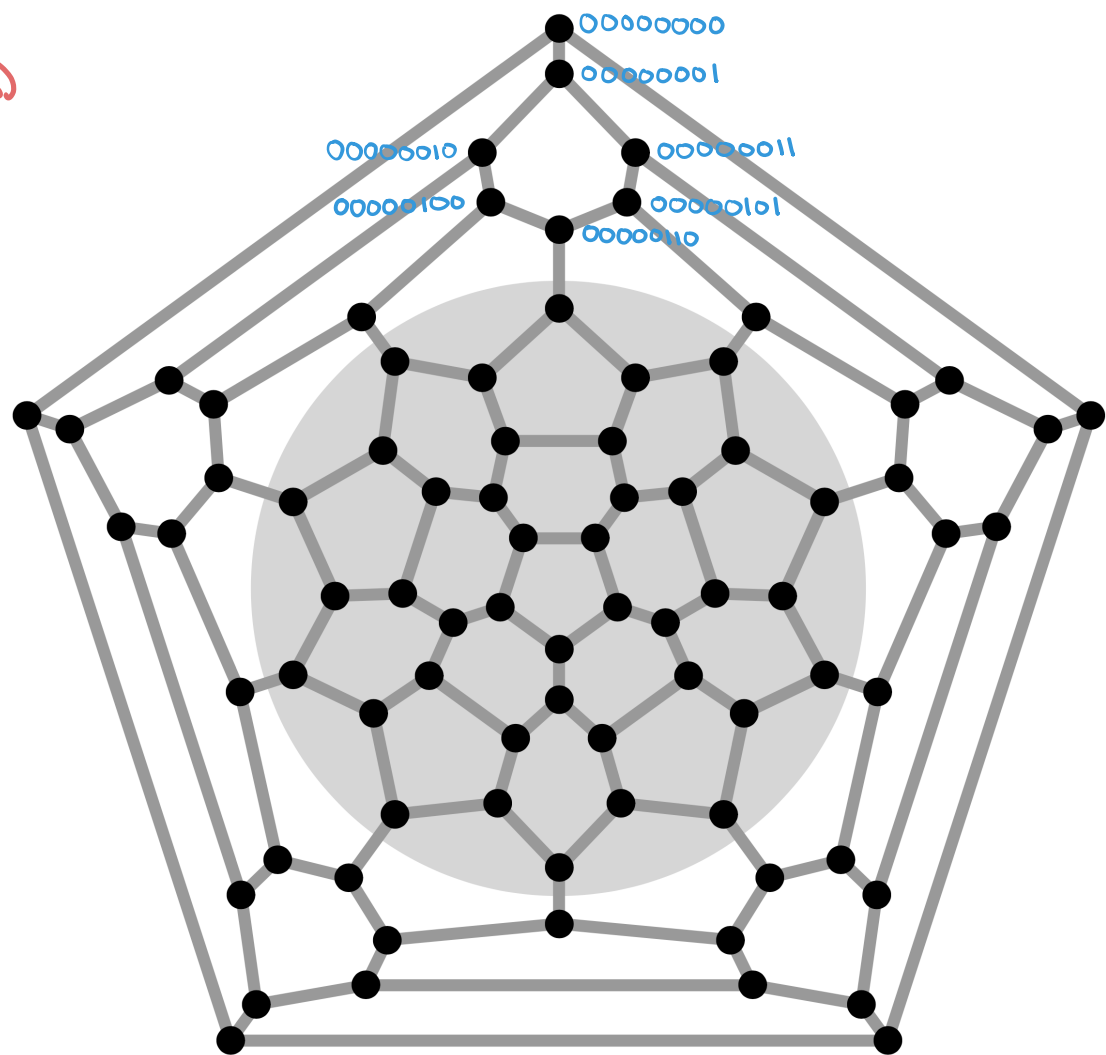
BPP is there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits



High level idea

- make constant degree expander $G=(V,E)$ w/ $V=\{0,1\}^m$
- take random walk $v_1, \dots, v_t \in \{0,1\}^m$ w/ $t=O(\log(1/\delta))$
- use each v_i as random bits for i th instance of algo

RP if there's a random. polytime algo that, on input x :
(randomized polynomial time)

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

BPP is there's a random. polytime algo that, on input x :
(bounded error, probabilistic polynomial time)

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits

Lemma "expander Chernoff"

G regular, undirected w/ spectral gap γ

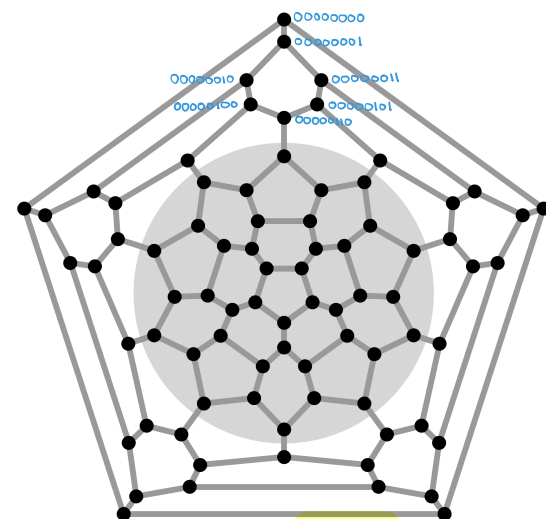
random walk $v_1, \dots, v_t$ (v_i unif. random)

$f: V \rightarrow [0,1]$, $\mu = E[f(v)]$ (w/ $v \in V$ unif. random)

for $\epsilon, \beta \in (0,1)$,

$$\Pr\left[\left|\frac{1}{k} \sum_{i=1}^k f(v_i) - \mu\right| \geq \epsilon\mu + \beta\right] \leq O(e^{\epsilon k}(\epsilon + \epsilon^2 - \beta))$$

$$\approx e^{\epsilon k(\epsilon - \beta)}$$



consider BPP (two-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

$$\frac{|B|}{|V|} \leq \frac{1}{3} \quad (\text{for example})$$

expander amplification fails iff at least half of random walk is in B

High level idea

- make constant degree expander $G=(V,E)$ w/ $V=\{0,1\}^m$
- take random walk $v_1, \dots, v_t \in \{0,1\}^m$ w/ $t = O(\log(1/\delta))$
- use each v_i as random bits for i th instance of algo

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

BPP is there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits

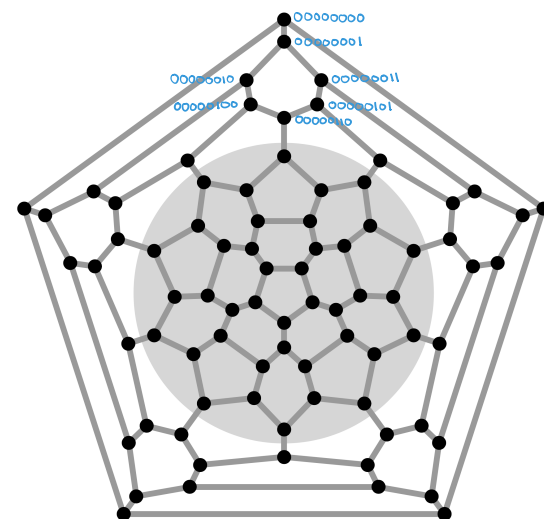
Lemma

G regular, undirected w/ spectral gap γ

$B \subset V$, $\mu = |B|/|V|$

random walk $v_1, \dots, v_t$ (v_1 unif. random)

$$P_r[v_1, \dots, v_t \in B] \leq (1 - (1 - \mu)\gamma)^t$$



consider RP (one-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

expander amplification
fails iff entire random
walk is in B

Lemma

G regular, undirected w/ spectral gap γ

$$B \subset V, \mu = |B|/|V|$$

random walk $v_1, \dots, v_t$ (v_1 unif. random)

$$P_r[v_1, \dots, v_t \in B] \leq (1 - (1 - \mu)\gamma)^t$$

Proof. let R = random walk matrix

$$\text{def } P: \mathbb{R}^V \rightarrow \mathbb{R}^V \text{ by } (Px)_v = \begin{cases} x_v & \text{if } v \in B \\ 0 & \text{otherwise} \end{cases}$$

$$\text{PRP} = \begin{cases} 1. \text{ drop mass outside } B \\ 2. \text{ take random step} \\ 3. \text{ drop mass outside } B \end{cases}$$

RP if there's a random. polytime algo that, on input x :
(randomized polynomial time)

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

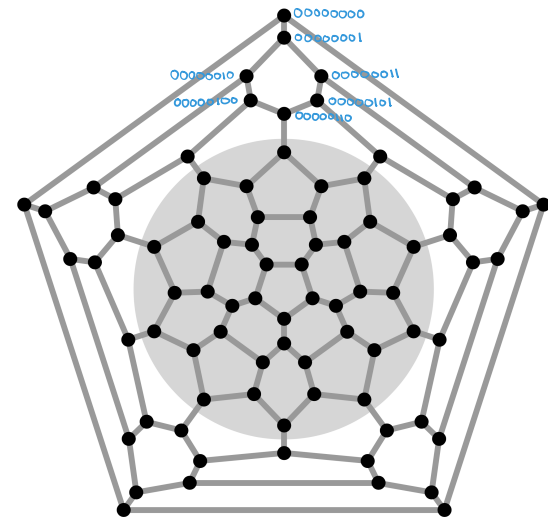
BPP if there's a random. polytime algo that, on input x :
(bounded error, probabilistic polynomial time)

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits



consider RP (one-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

expander amplification fails iff entire random walk is in B

$$(Px)_v = \begin{cases} x_v & \text{if } v \in B \\ 0 & \text{otherwise} \end{cases}$$

G regular, undirected w/ spectral gap γ
 $B \subset V$, $\mu = |B|/|V|$
 random walk $v_1, \dots, v_t$ (v_1 uniform random)

$$PRP = \begin{cases} 1. \text{ drop mass outside } B \\ 2. \text{ take random step} \\ 3. \text{ drop mass outside } B \end{cases}$$

$$\langle \mathbf{1}, (PRP)^\dagger(\mathbf{1}/n) \rangle = \text{prob. } v_1, \dots, v_t \in B$$

$$\langle \mathbf{1}, (PRP)^\dagger(\mathbf{1}/n) \rangle = \frac{\langle \mathbf{1}, (PRP)^\dagger \mathbf{1} \rangle}{\langle \mathbf{1}, \mathbf{1} \rangle} \leq \|PRP\|^\dagger$$

where $\|\cdot\|$ = operator norm = max abs. eigenvalue

$$\text{claim: } \|PRP\| \leq 1 - (1 - \mu)\gamma$$

(which completes the proof)

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

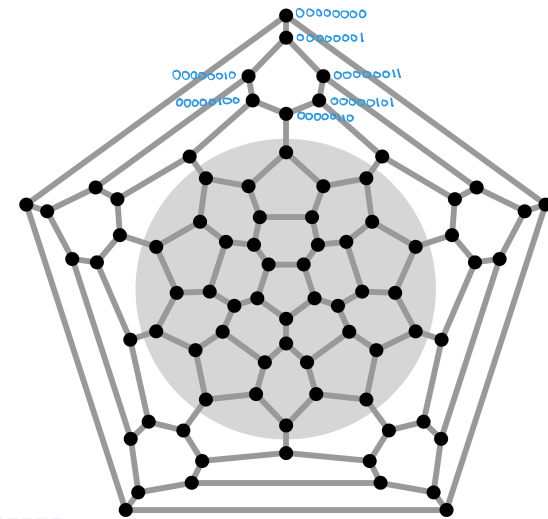
BPP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits



consider RP (one-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

expander amplification fails iff entire random walk is in B

claim: $\|PRP\| \leq 1 - (1 - \mu)\delta$

G regular $\Rightarrow R$ symmetric

$$R = \frac{1 \otimes 1}{n} + \mu_2(v_2 \otimes v_2) + \dots + \mu_n(v_n \otimes v_n)$$

$$= \frac{\delta}{n}(1 \otimes 1) + R' \quad \text{w/ } \|R'\| \leq 1 - \delta$$

PRP =

G regular w/ spec gap δ
 $B \subset V$, $\mu = |B|/|V|$

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

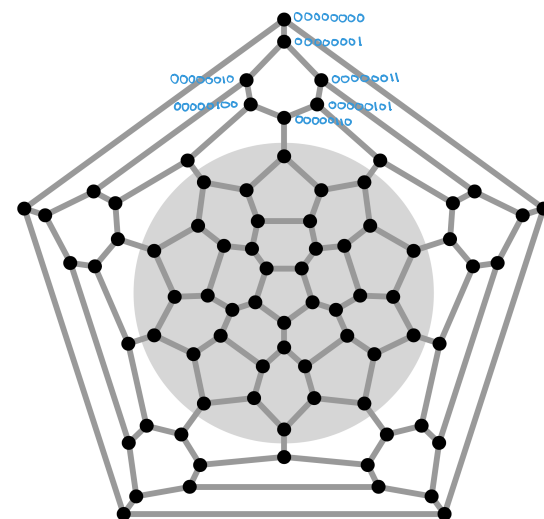
BPP is there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits



consider RP (one-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

expander amplification
 fails iff entire random
 walk is in B

G regular w/ spec gap γ
 $B \subset V$, $m = |B|/|V|$

RP if there's a random. polytime algo that, on input x :
(randomized polynomial time)

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

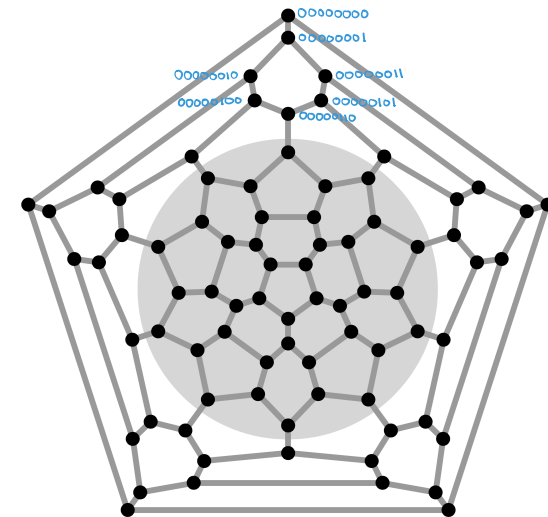
BPP if there's a random. polytime algo that, on input x :
(bounded error probabilistic polynomial time)

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits



consider RP (one-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

expander amplification
 fails iff entire random
 walk is in B

High level idea

- make constant degree expander $G=(V,E)$ w/ $V=\{0,1\}^m$
- take random walk $v_1, \dots, v_t \in \{0,1\}^m$ w/ $t=O(\log(1/\delta))$
- use each v_i as random bits for i th instance of algo

RP if there's a random. polytime algo that, on input x :
(randomized polynomial time)

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

BPP is there's a random. polytime algo that, on input x :
(bounded error, probabilistic polynomial time)

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits

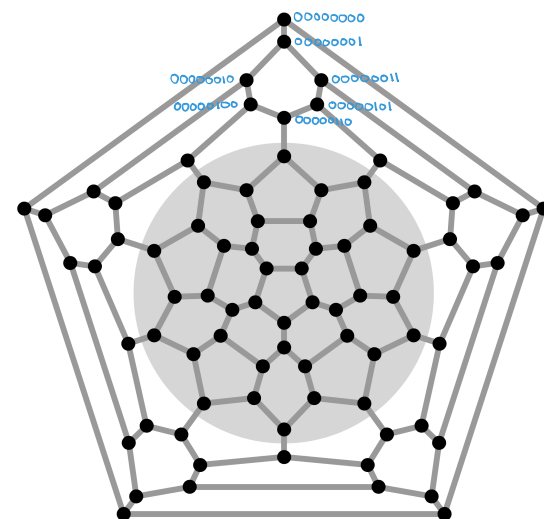
Lemma

G regular, undirected w/ spectral gap γ

$$B \subset V, \mu = |B|/|V|$$

random walk $v_1, \dots, v_t$ (v_1 unif. random)

$$P_r[v_1, \dots, v_t \in B] \leq (1 - (1 - \mu)\gamma)^t$$



consider RP (one-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

expander amplification

fails iff entire random

walk is in B

High level idea

- make constant degree expander $G=(V,E)$ w/ $V=\{0,1\}^m$
- take random walk $v_1, \dots, v_t \in \{0,1\}^m$ w/ $t=O(\log(1/\delta))$
- use each v_i as random bits for i th instance of algo

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

BPP is there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits

Lemma "expander Chernoff"

G regular, undirected w/ spectral gap $\gamma=1-\epsilon$

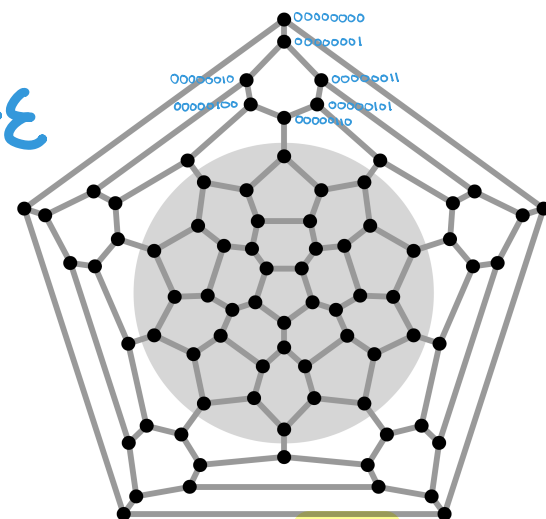
random walk $v_1, \dots, v_t$ (v_i unif. random)

$f: V \rightarrow [0,1]$, $\mu = E[f(v)]$ (w/ $v \in V$ unif. random)

for $\beta \in (0,1)$,

$$\Pr\left[\left|\frac{1}{k} \sum_{i=1}^k f(v_i) - \mu\right| \geq \epsilon\mu + \beta\right] \leq O(e^{\epsilon k}(\epsilon + \epsilon^2 - \beta))$$

$$\approx e^{\epsilon k(\epsilon - \beta)}$$



consider BPP (two-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

$$\frac{|B|}{|V|} \leq \frac{1}{3} \quad (\text{for example})$$

expander amplification fails iff at least half of random walk is in B

proof

G regular w/ spec gap γ
 $f: V \rightarrow [0,1]$, $\mu = E[f(v)]$
 $\epsilon, \beta \in (0,1)$

$$P[f(v_1) + \dots + f(v_k) \geq \lambda] \\ \leq E[e^{t(f(v_1) + \dots + f(v_k))}] e^{-t\lambda}$$

Let $F = \text{diag}(e^{tf(v_1)}, \dots, e^{tf(v_n)})$

claim: $E[e^{t(f(v_1) + \dots + f(v_n))}] = \frac{1}{n} \langle \mathbf{1}, F(RF)^k \mathbf{1} \rangle$

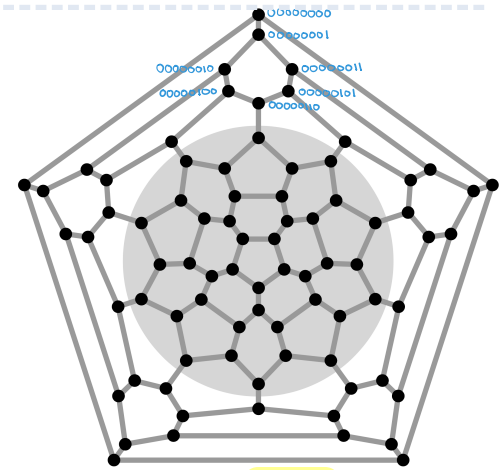
$FR \frac{1}{n} = E[\text{take a step, collect } e^{f(v)}]$

$(FR)^k \frac{1}{n} = E[\text{take } k \text{ steps, collect } e^{f(v)} \text{ each step}]$

RP if there's a random. polytime algo that, on input x :
(randomized polynomial time)
 • if $x \in L$, decides $x \in L$ w/ constant prob.
 • if $x \notin L$, always decides $x \notin L$

BPP is there's a random. polytime algo that, on input x :
(bounded error probabilistic polynomial time)
 • if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
 • if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem
 Given RP/BPP algo, one can get error prob δ w/
 • $O(\log(1/\delta))$ -factor increase in running time
 • $O(\log(1/\delta))$ additional random bits



consider BPP (two-sided error)
 call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)
 let $B = \{ \text{bad } v \}$
 $\frac{|B|}{|V|} \leq \frac{1}{3}$ (for example)
 expander amplification fails iff at least half of random walk is in B

$$P[f(v_1) + \dots + f(v_k) \geq \lambda]$$

$$\leq E[e^{t(f(v_1) + \dots + f(v_k))}] e^{-t\lambda}$$

$$\text{Let } F = \text{diag}(e^{tf(v_1)}, \dots, e^{tf(v_n)})$$

$$\checkmark E[e^{t(f(v_1) + \dots + f(v_n))}] = \frac{1}{n} \langle \mathbf{1}, F(RF)^k \mathbf{1} \rangle$$

$$\text{Let } R = \frac{\delta}{n}(\mathbf{1}\mathbf{1}^T) + R' \quad \text{w/ } \|R'\| \leq 1 - \delta = \varepsilon$$

$$\text{claim: } \|F^{1/2} R' F^{1/2}\| \leq \varepsilon e^t$$

$$f: V \rightarrow [0,1], \mu = E[f(v)]$$

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

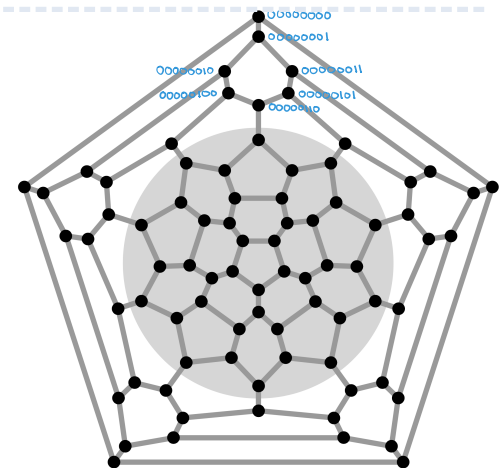
BPP is there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits



consider BPP (two-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{ \text{bad } v \}$

$$\frac{|B|}{|V|} \leq \frac{1}{3} \quad (\text{for example})$$

expander amplification fails iff at least half of random walk is in B

$$P[f(v_1) + \dots + f(v_k) \geq \lambda]$$

$$\leq E[e^{t(f(v_1) + \dots + f(v_k))}] e^{-t\lambda}$$

$$\text{Let } F = \text{diag}(e^{tf(v_1)}, \dots, e^{tf(v_n)})$$

$$\checkmark E[e^{t(f(v_1) + \dots + f(v_n))}] = \frac{1}{n} \langle \mathbf{1}, F(RF)^k \mathbf{1} \rangle$$

$$\text{Let } R = \frac{\delta}{n}(\mathbf{1} \otimes \mathbf{1}) + R' \quad \text{w/ } \|R'\| \leq 1 - \delta = \varepsilon$$

$$\checkmark \|F^{1/2} R' F^{1/2}\| \leq \varepsilon e^t$$

$$\text{claim: } \|F^{1/2}(\mathbf{1} \otimes \mathbf{1})F^{1/2}\| \leq E[e^{tf(v)}]$$

$$f: V \rightarrow [0,1], \mu = E[f(v)]$$

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

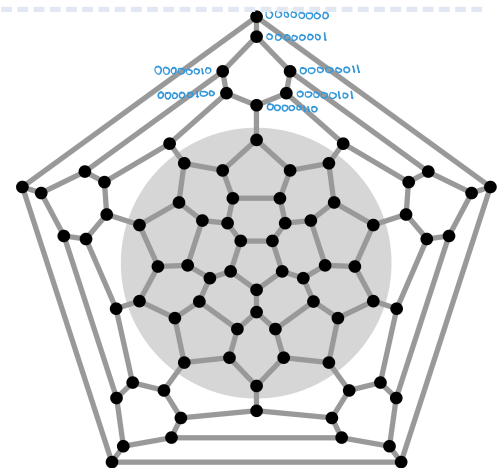
BPP is there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits



consider BPP (two-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{ \text{bad } v \}$

$$\frac{|B|}{|V|} \leq \frac{1}{3} \quad (\text{for example})$$

expander amplification fails iff at least half of random walk is in B

$$P[f(v_1) + \dots + f(v_k) \geq \lambda]$$

$$\leq E[e^{t(f(v_1) + \dots + f(v_k))}] e^{-t\lambda}$$

$$\text{Let } F = \text{diag}(e^{tf(v_1)}, \dots, e^{tf(v_n)})$$

$$\checkmark E[e^{t(f(v_1) + \dots + f(v_n))}] = \frac{1}{n} \langle \mathbf{1}, F(RF)^k \mathbf{1} \rangle$$

$$\text{Let } R = \frac{\gamma}{n}(\mathbf{1} \otimes \mathbf{1}) + R' \quad \text{w/ } \|R'\| \leq 1 - \gamma = \varepsilon$$

$$\checkmark \|F^{1/2} R' F^{1/2}\| \leq \varepsilon e^t$$

$$\checkmark \|F^{1/2}(\mathbf{1} \otimes \mathbf{1}) F^{1/2}\| \leq E[e^{tf(v)}]$$

$$\|F^{1/2} R F^{1/2}\| \leq$$

$f: V \rightarrow [0,1], \mu = E[f(v)]$

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

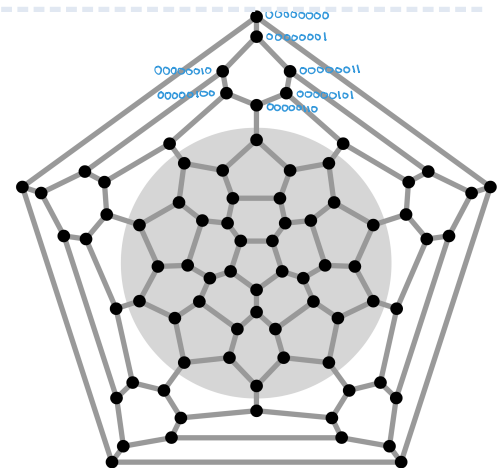
BPP is there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits



consider BPP (two-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

$$\frac{|B|}{|V|} \leq \frac{1}{3} \quad (\text{for example})$$

expander amplification fails iff at least half of random walk is in B

$$P[f(v_1) + \dots + f(v_k) \geq \lambda]$$

$$\leq E[e^{t(f(v_1) + \dots + f(v_k))}] e^{-t\lambda}$$

$$\text{Let } F = \text{diag}(e^{tf(v_1)}, \dots, e^{tf(v_n)})$$

$$\checkmark E[e^{t(f(v_1) + \dots + f(v_n))}] = \frac{1}{n} \langle \mathbf{1}, F(RF)^k \mathbf{1} \rangle$$

$$\text{Let } R = \frac{\gamma}{n}(\mathbf{1} \otimes \mathbf{1}) + R' \quad \text{w/ } \|R'\| \leq 1 - \gamma = \varepsilon$$

$$\checkmark \|F^{1/2} R' F^{1/2}\| \leq \varepsilon e^t$$

$$\checkmark \|F^{1/2}(\mathbf{1} \otimes \mathbf{1}) F^{1/2}\| \leq E[e^{tf(v)}]$$

$$\checkmark \|F^{1/2} R F^{1/2}\| \leq e^{(t+t^2)(\varepsilon + \mu)}$$

$f: V \rightarrow [0,1], \mu = E[f(v)]$

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

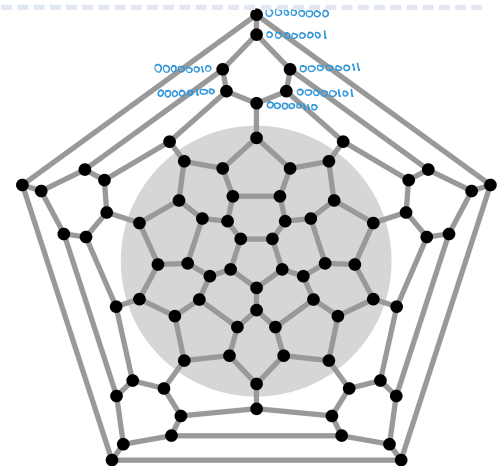
BPP is there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits



consider BPP (two-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

$$\frac{|B|}{|V|} \leq \frac{1}{3} \quad (\text{for example})$$

expander amplification fails iff at least half of random walk is in B

$$P[f(v_1) + \dots + f(v_k) \geq \lambda]$$

$$\leq E[e^{t(f(v_1) + \dots + f(v_k))}] e^{-t\lambda}$$

Let $F = \text{diag}(e^{tf(v_1)}, \dots, e^{tf(v_n)})$

✓ $E[e^{t(f(v_1) + \dots + f(v_n))}] = \frac{1}{n} \langle \mathbf{1}, F(RF)^k \mathbf{1} \rangle$

✓ $\|F^{1/2} R F^{1/2}\| \leq e^{(t+t^2)(\epsilon + \mu)}$

$$\frac{1}{n} \langle \mathbf{1}, F(RF)^k \mathbf{1} \rangle \leq$$

$f: V \rightarrow [0,1], \mu = E[f(v)]$

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

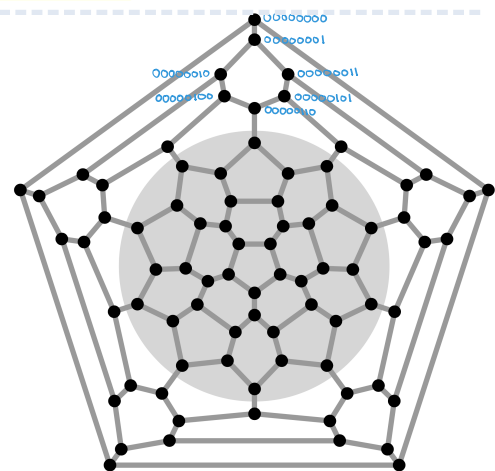
BPP is there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits



consider BPP (two-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{ \text{bad } v \}$

$$\frac{|B|}{|V|} \leq \frac{1}{3} \quad (\text{for example})$$

expander amplification fails iff at least half of random walk is in B

RP if there's a random. polytime algo that, on input x :
(randomized polynomial time)

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

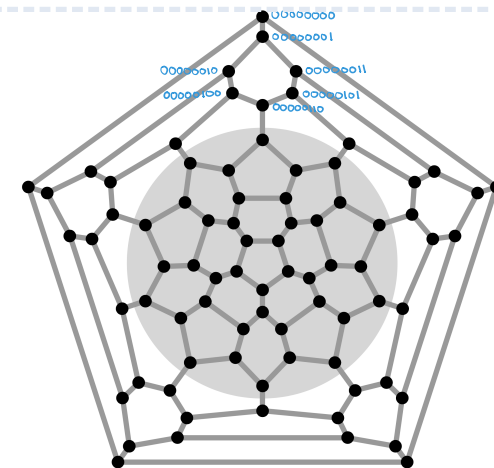
BPP is there's a random. polytime algo that, on input x :
(bounded error probabilistic polynomial time)

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits



consider BPP (two-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

$$\frac{|B|}{|V|} \leq \frac{1}{3} \quad (\text{for example})$$

expander amplification
fails iff at least half of
random walk is in B

High level idea

- make constant degree expander $G=(V,E)$ w/ $V=\{0,1\}^m$
- take random walk $v_1, \dots, v_t \in \{0,1\}^m$ w/ $t=O(\log(1/\delta))$
- use each v_i as random bits for i th instance of algo

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

BPP is there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits

✓ Lemma "expander Chernoff"

G regular, undirected w/ spectral gap $\gamma=1-\epsilon$

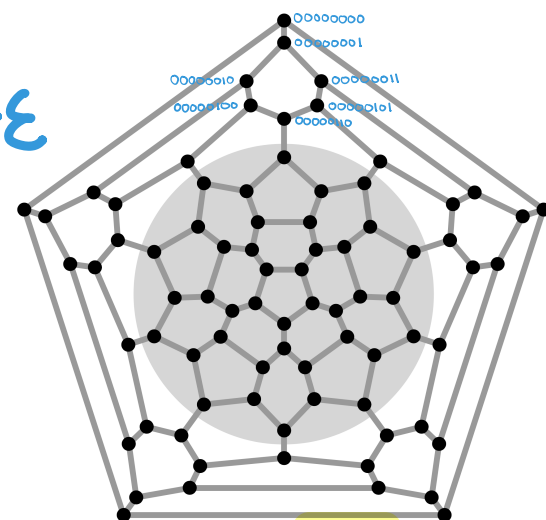
random walk $v_1, \dots, v_t$ (v_i unif. random)

$f: V \rightarrow [0,1]$, $\mu = E[f(v)]$ (w/ $v \in V$ unif. random)

for $\beta \in (0,1)$,

$$\Pr\left[\left|\frac{1}{k} \sum_{i=1}^k f(v_i) - \mu\right| \geq \epsilon\mu + \beta\right] \leq O(e^{\epsilon k}(\epsilon + \epsilon^2 - \beta))$$

$$\approx e^{\epsilon k(\epsilon - \beta)}$$



consider BPP (two-sided error)

call $v \in \{0,1\}^m$ "bad" if algo makes error w/ v (as random bits)

let $B = \{\text{bad } v\}$

$$\frac{|B|}{|V|} \leq \frac{1}{3} \quad (\text{for example})$$

expander amplification fails iff at least half of random walk is in B

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

BPP if there's a random. polytime algo that, on input x :

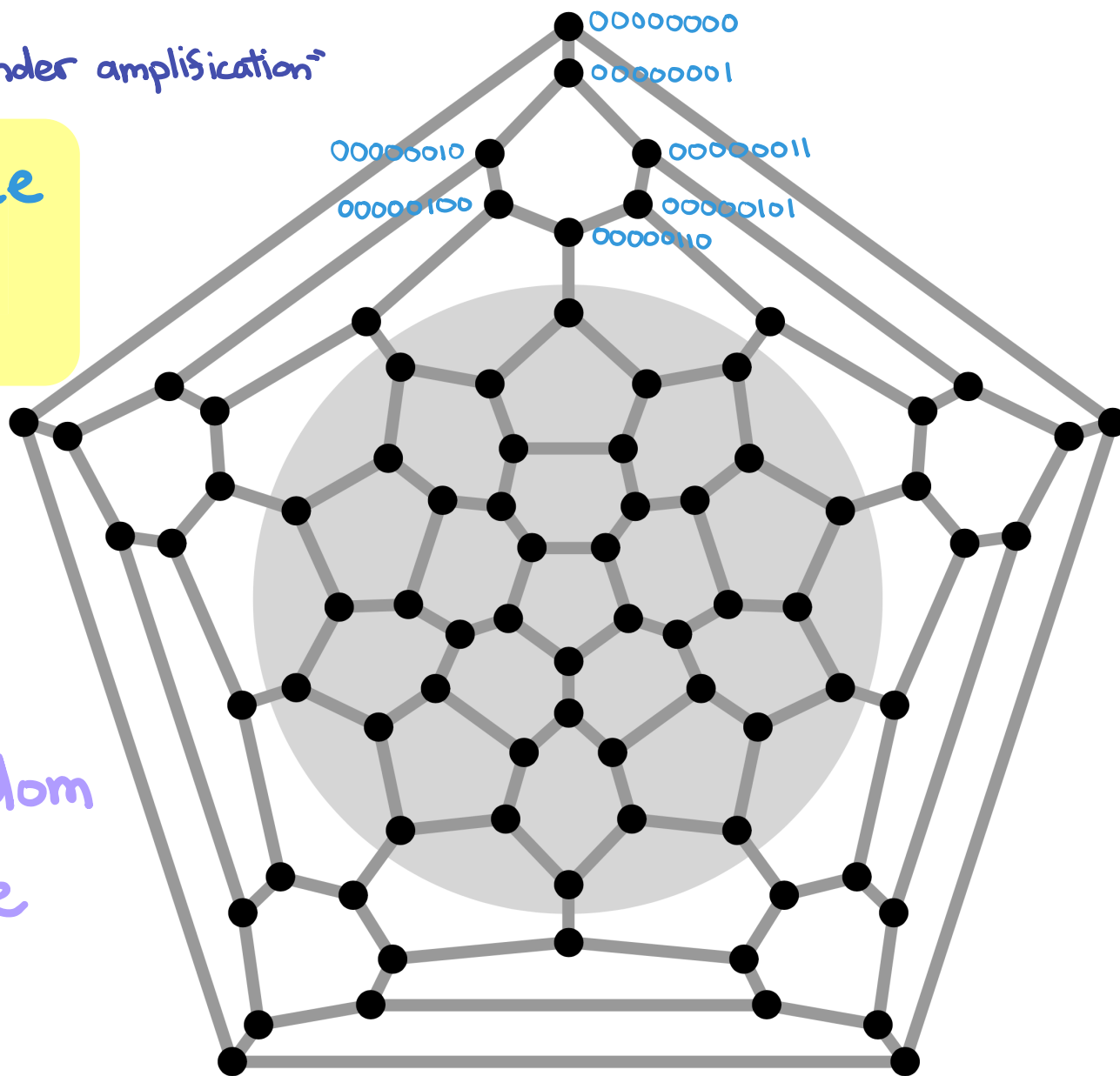
- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

High level idea "expander amplification"

- make constant degree expander $G=(V,E)$ w/ $V = \{0,1\}^m$

- take random walk $v_1, \dots, v_t \in \{0,1\}^m$ w/ $t = O(\log(1/\delta))$

- use each v_i as random bits for i th instance of algo



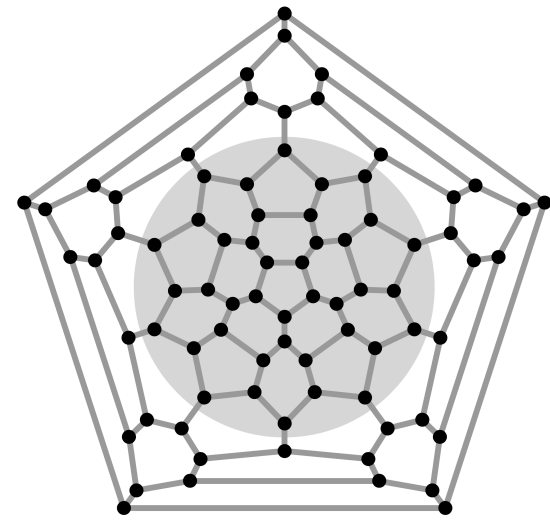
let H be constant degree expander
w/ degree d , d^8 vertices, $\gamma \geq 7/8$

$$G_1 = H^2$$

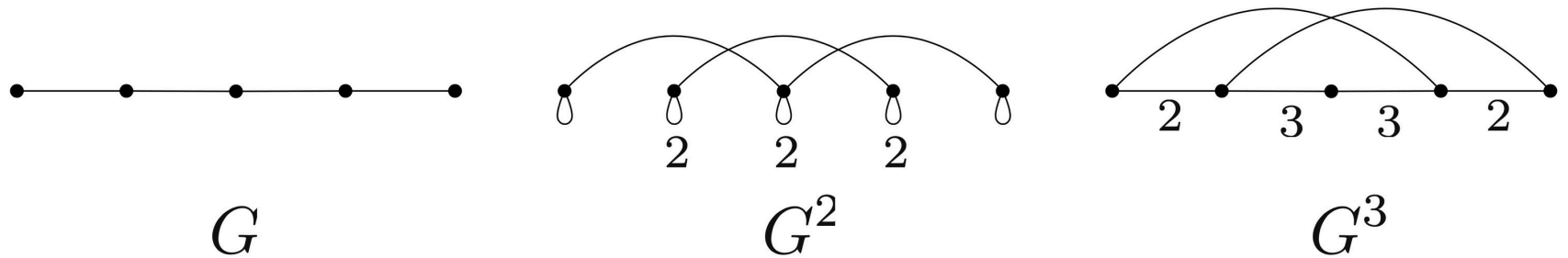
graph tensor product

$$G_{t+1} = Z((G_t \otimes G_t) \mid H)$$

zig-zag product



Powering



$G^k =$ graph w/ an edge for every k step walk in G
(w/ repetition)

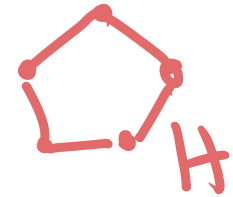
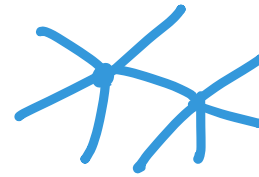
random walk R^k has eigenvalues $1, \mu_2^k, \dots, \mu_n^k$

$$\delta(G^k) = 1 - (1 - \delta(G))^k \quad \text{😊}$$

degree increases from d to d^k 😞

Zig-zag product $Z(G, H)$

let G be d -regular

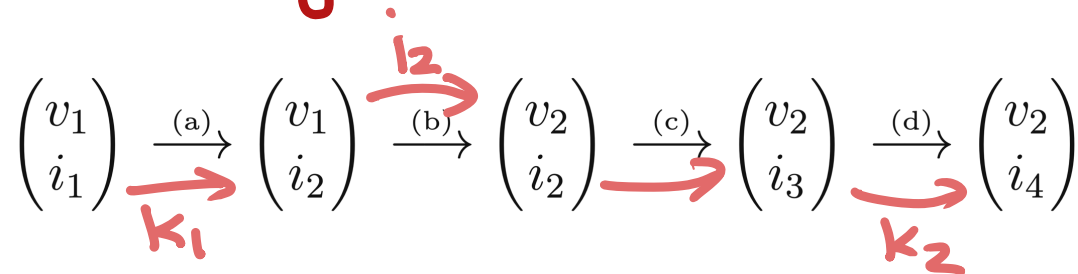


H be d_0 -regular w/ d_0 vertices (H will be $O(D)$ -degree expander)

vertices: $V_G \times V_H = \{(u, v) : u \in V_G, v \in V_H\}$

edges: for $(v_1, i_1) \in V_G \times V_H$, $k_1, k_2 \in [d_0]$:

the (k_1, k_2) th neighbor of (v_1, i_1) is (v_2, i_4) where:



where:

- (a) is a step in H from i_1 to its k_1 th neighbor, i_2 .
- (b) moves v_1 to the i_2 th outgoing neighbor of v_1 , v_2 .
- (c) moves from i_2 to i_3 where v_1 is the i_3 th neighbor of v_2 .
- (d) moves i_3 to its k_2 th neighbor in H .

(so $Z(G, H)$ is d_0^2 -regular)

Lemma: $Z(G|H)$ has

- n vertices
- degree d_0^2 😊
- spectral gap $\gamma_G \gamma_H^2$ 😊

Zig-zag product $Z(G,H)$

let G be d -regular

H be d_0 -regular w/ d vertices (H will be $O(1)$ -degree expander)

vertices: $V_G \times V_H = \{(u,v) : u \in V_G, v \in V_H\}$

edges: for $(v_1, i_1) \in V_G \times V_H$, $k_1, k_2 \in [d_0]$:

the (k_1, k_2) th neighbor of (v_1, i_1) is (v_2, i_4) where:

$$\begin{pmatrix} v_1 \\ i_1 \end{pmatrix} \xrightarrow{(a)} \begin{pmatrix} v_1 \\ i_2 \end{pmatrix} \xrightarrow{(b)} \begin{pmatrix} v_2 \\ i_2 \end{pmatrix} \xrightarrow{(c)} \begin{pmatrix} v_2 \\ i_3 \end{pmatrix} \xrightarrow{(d)} \begin{pmatrix} v_2 \\ i_4 \end{pmatrix}$$

where:

- (a) is a step in H from i_1 to its k_1 th neighbor, i_2 .
- (b) moves v_1 to the i_2 th outgoing neighbor of v_1 , v_2 .
- (c) moves from i_2 to i_3 where v_1 is the i_3 th neighbor of v_2 .
- (d) moves i_3 to its k_2 th neighbor in H .

(so $Z(G,H)$ is d_0^2 -regular)

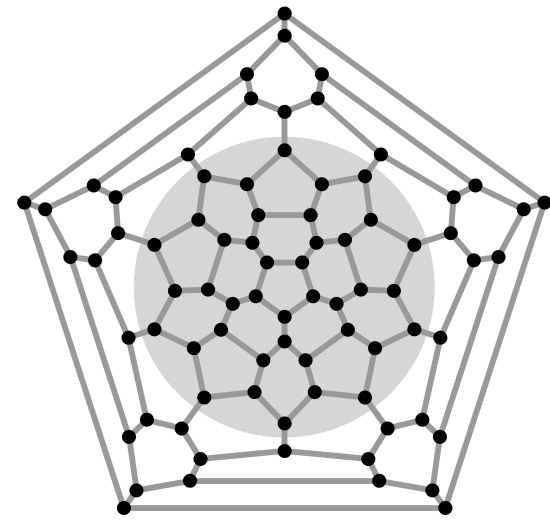
let H be constant degree expander
w/ degree d , d^8 vertices, $\gamma \geq 7/8$

$$G_1 = H^2$$

$$G_{t+1} = Z((G_t \otimes G_t) \mid H)$$

zig-zag product

graph tensor product



Tensor products

for $A \in \mathbb{R}^{m \times m}$, $B \in \mathbb{R}^{n \times n}$,

$A \otimes B$ is a linear map on $\mathbb{R}^{m \times n}$ defined by

$$(A \otimes B)C = ACB \quad \text{for } C \in \mathbb{R}^{m \times n}$$

special case:

$$(A \otimes B)(c \otimes d) = (Ac) \otimes (Bd) \quad c \in \mathbb{R}^m, d \in \mathbb{R}^n$$

Fact: $A \otimes B$ has eigenvalues $\lambda_A \lambda_B$ where

λ_A is an eigenvalue of A ,

λ_B B

Tensor product $I \otimes R_H$

$I \otimes R_H$ is a linear map on $\mathbb{R}^{V \times d}$ defined by

$$((I \otimes R_H)X)_v = R_H X_v \quad X \in \mathbb{R}^{V \times d}$$

where $X_v = v$ th row of X .

Fact: $I \otimes (A+B) = I \otimes A + I \otimes B$

Graph tensor

$$\text{let } G_1 = (V_1, E_1), \quad G_2 = (V_2, E_2)$$

$$G_1 \otimes G_2 = (V_1 \times V_2, E_1 \times E_2)$$

$$V_1 \times V_2 = \{(v_1, v_2) : v_1 \in V_1, v_2 \in V_2\}$$

$$E_1 \times E_2 = \{((u_1, u_2), (v_1, v_2)) : (u_1, v_1) \in E_1, (u_2, v_2) \in E_2\}$$

is $R_1 =$ random walk for G_1

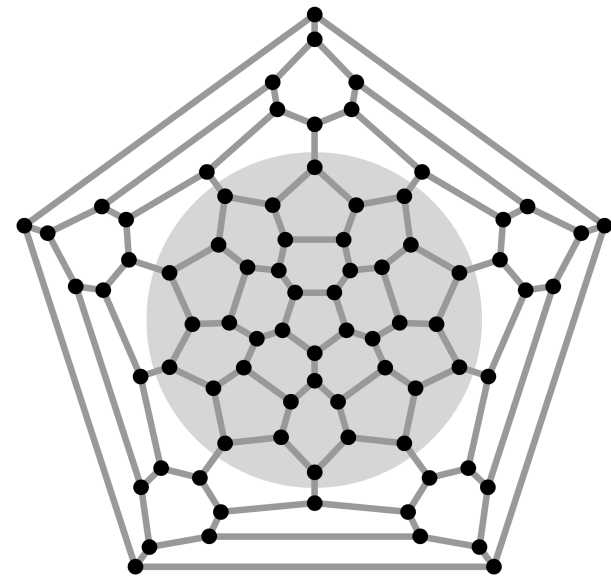
$R_2 =$    G_2

then $R_1 \otimes R_2 =$ random walk for $G_1 \otimes G_2$

let H be constant degree expander
w/ degree d , d^8 vertices, $\gamma \geq 7/8$

$$G_1 = H^2$$

$$G_{t+1} = Z((G_t \otimes G_t) | H)$$



G^k = graph w/ edge for every k step walk
(w/ repetition)

$$\gamma(G^k) = 1 - (1 - \gamma(G))^k$$

degree increases from d to d^k

graph	G	$G \otimes G$	$(G \otimes G)^2$	$Z((G \otimes G)^2 H)$
vertices	n			
degree	d^2			
γ	γ			

Lemma: $Z(G | H)$ has

- $\bullet$ n vertices
- $\bullet$ degree d^2
- $\bullet$ spectral gap $\gamma_G \gamma_H^2$

Fact: $A \otimes B$ has eigenvalues $\lambda_A \lambda_B$
w/ λ_A is an eigenvalue of A ,
 λ_B is an eigenvalue of B

$$\gamma \geq \frac{1}{2} \Rightarrow$$

RP if there's a random. polytime algo that, on input x :
(randomized polynomial time)

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

BPP if there's a random. polytime algo that, on input x :
(bounded error probabilistic polynomial time)

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits

ideas?

Theorem

Given RP/BPP algo, one can get error prob δ w/

- $O(\log(1/\delta))$ -factor increase in running time
- $O(\log(1/\delta))$ additional random bits

RP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ constant prob.
- if $x \notin L$, always decides $x \notin L$

BPP if there's a random. polytime algo that, on input x :

- if $x \in L$, decides $x \in L$ w/ prob $\geq 2/3$
- if $x \notin L$, decides $x \notin L$ w/ prob. $\geq 2/3$

High level idea "expander amplification"

- make constant degree expander $G=(V,E)$ w/ $V = \{0,1\}^m$
- take random walk $v_1, \dots, v_t \in \{0,1\}^m$ w/ $t = O(\log(1/\delta))$
- use each v_i as random bits for i th instance of algo

