Today.

Markov chain mixing analysis.

Today.

Markov chain mixing analysis.

Perhaps: reducing the use of randomness.

Sampling.

Sampling: Random element of subset $S \subset \{0,1\}^n$ or $\{0,\ldots,k\}^k$.

Sampling.

Sampling: Random element of subset $S \subset \{0,1\}^n$ or $\{0,\ldots,k\}^k$.

Related Problem: Approximate $|S|$ within factor of $1+\varepsilon$.

Sampling.

Sampling: Random element of subset $S \subset \{0,1\}^n$ or $\{0,\ldots,k\}^k$.

Related Problem: Approximate $|S|$ within factor of $1+\varepsilon$.

Random walk to do both for some interesting sets S.

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Example: P defined by set of linear inequalities.

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Example: P defined by set of linear inequalities.
Or other "membership oracle" for P

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Example: P defined by set of linear inequalities.
Or other "membership oracle" for P

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Example: P defined by set of linear inequalities.
Or other "membership oracle" for P

S is set of grid points inside Convex Body.

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Example: P defined by set of linear inequalities.
Or other "membership oracle" for P

S is set of grid points inside Convex Body.
Grid points that satisfy linear inequalities.

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Example: P defined by set of linear inequalities.
Or other “membership oracle” for P

S is set of grid points inside Convex Body.
Grid points that satisfy linear inequalities.
or “other” membership oracle.

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Example: P defined by set of linear inequalities.
Or other "membership oracle" for P

S is set of grid points inside Convex Body.
Grid points that satisfy linear inequalities.
or "other" membership oracle.
Choose a uniformly random elt?

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Example: P defined by set of linear inequalities.
Or other "membership oracle" for P

S is set of grid points inside Convex Body.
Grid points that satisfy linear inequalities.
or "other" membership oracle.
Choose a uniformly random elt?

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Example: P defined by set of linear inequalities.
Or other "membership oracle" for P

S is set of grid points inside Convex Body.
Grid points that satisfy linear inequalities.
or "other" membership oracle.
Choose a uniformly random elt?
Easy to choose randomly from $[k]^n$ which is big.

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Example: P defined by set of linear inequalities.
Or other "membership oracle" for P

S is set of grid points inside Convex Body.
Grid points that satisfy linear inequalities.
or "other" membership oracle.
Choose a uniformly random elt?
Easy to choose randomly from $[k]^n$ which is big.

For convex body?

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Example: P defined by set of linear inequalities.
Or other "membership oracle" for P

S is set of grid points inside Convex Body.
Grid points that satisfy linear inequalities.
or "other" membership oracle.

Choose a uniformly random elt?
Easy to choose randomly from $[k]^n$ which is big.

For convex body?
Choose random point in $[k]^n$ and check if in P.

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Example: P defined by set of linear inequalities.
Or other "membership oracle" for P

S is set of grid points inside Convex Body.
Grid points that satisfy linear inequalities.
or "other" membership oracle.
Choose a uniformly random elt?
Easy to choose randomly from $[k]^n$ which is big.

For convex body?
Choose random point in $[k]^n$ and check if in P.
Works.

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Example: P defined by set of linear inequalities.
Or other "membership oracle" for P

S is set of grid points inside Convex Body.
Grid points that satisfy linear inequalities.
or "other" membership oracle.
Choose a uniformly random elt?
Easy to choose randomly from $[k]^n$ which is big.

For convex body?
Choose random point in $[k]^n$ and check if in P.
Works.
But P could be exponentially small compared to $|[k]^n|$.

Convex Bodies.

$S \subset [k]^n$ is grid points inside Convex Body.

Ex: Numerically integrate convex function in d dimensions.
Compute $\sum_i v_i Vol(f(x) > v_i)$ where $v_i = i\delta$.

Example: P defined by set of linear inequalities.
Or other "membership oracle" for P

S is set of grid points inside Convex Body.
Grid points that satisfy linear inequalities.
or "other" membership oracle.
Choose a uniformly random elt?
Easy to choose randomly from $[k]^n$ which is big.

For convex body?
Choose random point in $[k]^n$ and check if in P.
Works.
But P could be exponentially small compared to $|[k]^n|$.
Takes a long time to even find a point in P.

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)

Degree: $2d$.

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph?

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph? Big!

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph? Big!
So big it ..

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph? Big!
So big it ..it

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph? Big!
So big it ..it INSERT JOKE HERE.

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph? Big!
So big it ..it INSERT JOKE HERE.
$O(k^n)$ if coordinates in $[k]$.

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph? Big!
So big it ..it INSERT JOKE HERE.
$O(k^n)$ if coordinates in $[k]$.
That's a big graph!

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph? Big!
So big it ..it INSERT JOKE HERE.
$O(k^n)$ if coordinates in $[k]$.
That's a big graph!

How to find a random node?

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph? Big!
So big it ..it INSERT JOKE HERE.
$O(k^n)$ if coordinates in $[k]$.
That's a big graph!

How to find a random node?

Start at a grid point, and take a (random) walk.

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph? Big!
So big it ..it INSERT JOKE HERE.
$O(k^n)$ if coordinates in $[k]$.
That's a big graph!

How to find a random node?

Start at a grid point, and take a (random) walk.

When close to uniform distribution...

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph? Big!
So big it ..it INSERT JOKE HERE.
$O(k^n)$ if coordinates in $[k]$.
That's a big graph!

How to find a random node?

Start at a grid point, and take a (random) walk.

When close to uniform distribution...have a sample point.

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph? Big!
So big it ..it INSERT JOKE HERE.
$O(k^n)$ if coordinates in $[k]$.
That's a big graph!

How to find a random node?

Start at a grid point, and take a (random) walk.

When close to uniform distribution...have a sample point.

How long does this take?

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph? Big!
So big it ..it INSERT JOKE HERE.
$O(k^n)$ if coordinates in $[k]$.
That's a big graph!

How to find a random node?

Start at a grid point, and take a (random) walk.

When close to uniform distribution...have a sample point.

How long does this take? More later.

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph? Big!
So big it ..it INSERT JOKE HERE.
$O(k^n)$ if coordinates in $[k]$.
That's a big graph!

How to find a random node?

Start at a grid point, and take a (random) walk.

When close to uniform distribution...have a sample point.

How long does this take? More later.
But remember power method...

Convex Body Graph.

$S \subset [k]^n$ is set of grid points inside Convex Body.

Sample Space: S.

Graph on grid points inside P or on Sample Space.

One neighbor in each direction for each dimension
(if neighbor is inside P.)
Degree: $2d$.

How big is graph? Big!
So big it ..it INSERT JOKE HERE.
$O(k^n)$ if coordinates in $[k]$.
That's a big graph!

How to find a random node?

Start at a grid point, and take a (random) walk.

When close to uniform distribution...have a sample point.

How long does this take? More later.
But remember power method...which finds first eigenvector.

Spanning Trees.

Problem: How many?

Spanning Trees.

Problem: How many?

Another Problem: find a random one.

Spanning Trees.

Problem: How many?

Another Problem: find a random one.

Algorithm:

Spanning Trees.

Problem: How many?

Another Problem: find a random one.

Algorithm:
  Start with spanning tree.

Spanning Trees.

Problem: How many?

Another Problem: find a random one.

Algorithm:
 Start with spanning tree.
 Repeat:

Spanning Trees.

Problem: How many?

Another Problem: find a random one.

Algorithm:
  Start with spanning tree.
  Repeat:
    Swap a random nontree edge with a random tree edge.

Spanning Trees.

Problem: How many?

Another Problem: find a random one.

Algorithm:
  Start with spanning tree.
  Repeat:
    Swap a random nontree edge with a random tree edge.

Spanning Trees.

Problem: How many?

Another Problem: find a random one.

Algorithm:
  Start with spanning tree.
  Repeat:
    Swap a random nontree edge with a random tree edge.

How long?

Spanning Trees.

Problem: How many?

Another Problem: find a random one.

Algorithm:
  Start with spanning tree.
  Repeat:
    Swap a random nontree edge with a random tree edge.

How long?

Sample space graph (BIG GRAPH) of spanning trees.

Spanning Trees.

Problem: How many?

Another Problem: find a random one.

Algorithm:
  Start with spanning tree.
  Repeat:
    Swap a random nontree edge with a random tree edge.

How long?

Sample space graph (BIG GRAPH) of spanning trees.
  Node for each tree.

Spanning Trees.

Problem: How many?

Another Problem: find a random one.

Algorithm:
  Start with spanning tree.
  Repeat:
    Swap a random nontree edge with a random tree edge.

How long?

Sample space graph (BIG GRAPH) of spanning trees.
  Node for each tree.
  Neighboring trees differ in two edges.

Spanning Trees.

Problem: How many?

Another Problem: find a random one.

Algorithm:
  Start with spanning tree.
  Repeat:
    Swap a random nontree edge with a random tree edge.

How long?

Sample space graph (BIG GRAPH) of spanning trees.
  Node for each tree.
  Neighboring trees differ in two edges.

Algorithm is random walk on BIG GRAPH (sample space graph.)

Spin systems.

Each element of S may have associated weight.

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?
2 or 3 dimensional grid of particles.

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?
2 or 3 dimensional grid of particles.
Particle State ± 1.

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?
2 or 3 dimensional grid of particles.
Particle State ± 1. System State $\{-1, +1\}^n$.

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?
2 or 3 dimensional grid of particles.
Particle State ± 1. System State $\{-1,+1\}^n$.

Energy on local interactions: $E = \sum_{(i,j)} -\sigma_i \sigma_j$.

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?
2 or 3 dimensional grid of particles.
Particle State ± 1. System State $\{-1,+1\}^n$.

Energy on local interactions: $E = \sum_{(i,j)} -\sigma_i \sigma_j$.

"Ferromagnetic regime": same spin is good.

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?
2 or 3 dimensional grid of particles.
Particle State ± 1. System State $\{-1,+1\}^n$.

Energy on local interactions: $E = \sum_{(i,j)} -\sigma_i \sigma_j$.

"Ferromagnetic regime": same spin is good.

Gibbs distribution $\propto e^{-E/kT}$.

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?
2 or 3 dimensional grid of particles.
Particle State ± 1. System State $\{-1, +1\}^n$.

Energy on local interactions: $E = \sum_{(i,j)} -\sigma_i \sigma_j$.

"Ferromagnetic regime": same spin is good.

Gibbs distribution $\propto e^{-E/kT}$.

Physical properties from Gibbs distribution.

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?
  2 or 3 dimensional grid of particles.
  Particle State ± 1. System State $\{-1, +1\}^n$.

  Energy on local interactions: $E = \sum_{(i,j)} -\sigma_i \sigma_j$.

"Ferromagnetic regime": same spin is good.

Gibbs distribution $\propto e^{-E/kT}$.

  Physical properties from Gibbs distribution.

Metropolis Algorithm:

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?

2 or 3 dimensional grid of particles.

Particle State ± 1. System State $\{-1,+1\}^n$.

Energy on local interactions: $E = \sum_{(i,j)} -\sigma_i \sigma_j$.

"Ferromagnetic regime": same spin is good.

Gibbs distribution $\propto e^{-E/kT}$.

Physical properties from Gibbs distribution.

Metropolis Algorithm:

At x, generate y with a single random flip.

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?
 2 or 3 dimensional grid of particles.
 Particle State ± 1. System State $\{-1,+1\}^n$.

 Energy on local interactions: $E = \sum_{(i,j)} -\sigma_i \sigma_j$.

"Ferromagnetic regime": same spin is good.

Gibbs distribution $\propto e^{-E/kT}$.

 Physical properties from Gibbs distribution.

Metropolis Algorithm:
 At x, generate y with a single random flip.
 Go to y with probability $\min(1, w(y)/w(x))$

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?
2 or 3 dimensional grid of particles.
Particle State ± 1. System State $\{-1,+1\}^n$.

Energy on local interactions: $E = \sum_{(i,j)} -\sigma_i \sigma_j$.

"Ferromagnetic regime": same spin is good.

Gibbs distribution $\propto e^{-E/kT}$.

Physical properties from Gibbs distribution.

Metropolis Algorithm:
At x, generate y with a single random flip.
Go to y with probability $\min(1, w(y)/w(x))$

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?
 2 or 3 dimensional grid of particles.
 Particle State ± 1. System State $\{-1,+1\}^n$.

 Energy on local interactions: $E = \sum_{(i,j)} -\sigma_i \sigma_j$.

"Ferromagnetic regime": same spin is good.

Gibbs distribution $\propto e^{-E/kT}$.

 Physical properties from Gibbs distribution.

Metropolis Algorithm:
 At x, generate y with a single random flip.
 Go to y with probability $\min(1, w(y)/w(x))$

Random walk in sample space graph (BIG GRAPH ALERT)

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?
2 or 3 dimensional grid of particles.
Particle State ± 1. System State $\{-1,+1\}^n$.

Energy on local interactions: $E = \sum_{(i,j)} -\sigma_i \sigma_j$.

"Ferromagnetic regime": same spin is good.

Gibbs distribution $\propto e^{-E/kT}$.

Physical properties from Gibbs distribution.

Metropolis Algorithm:
At x, generate y with a single random flip.
Go to y with probability $\min(1, w(y)/w(x))$

Random walk in sample space graph (BIG GRAPH ALERT)
(not random walk in 2d grid of particles.)

Spin systems.

Each element of S may have associated weight.

Sample element proportional to weight.

Example?
2 or 3 dimensional grid of particles.
Particle State ± 1. System State $\{-1, +1\}^n$.

Energy on local interactions: $E = \sum_{(i,j)} -\sigma_i \sigma_j$.

"Ferromagnetic regime": same spin is good.

Gibbs distribution $\propto e^{-E/kT}$.

Physical properties from Gibbs distribution.

Metropolis Algorithm:
At x, generate y with a single random flip.
Go to y with probability $\min(1, w(y)/w(x))$

Random walk in sample space graph (BIG GRAPH ALERT)
(not random walk in 2d grid of particles.)

Markov Chain on statespace of system.

Sampling structures and the BIG GRAPH

Sampling Algorithms $\equiv$ Random walk on BIG GRAPH.

Sampling structures and the BIG GRAPH

Sampling Algorithms $\equiv$ Random walk on BIG GRAPH. Small degree.

Vertices — Grid points in convex body.

Neighbors

Degree (ish)

Sampling structures and the BIG GRAPH

Sampling Algorithms $\equiv$ Random walk on BIG GRAPH. Small degree.

Vertices — Grid points in convex body.

Neighbors — Change one dimension

Degree (ish)

Sampling structures and the BIG GRAPH

Sampling Algorithms $\equiv$ Random walk on BIG GRAPH. Small degree.

Vertices	Neighbors	Degree (ish)
Grid points in convex body.	Change one dimension	$2d$
Spanning Trees.		

Sampling structures and the BIG GRAPH

Sampling Algorithms $\equiv$ Random walk on BIG GRAPH. Small degree.

Vertices	Neighbors	Degree (ish)
Grid points in convex body.	Change one dimension	$2d$
Spanning Trees.	Change two edges.	

Sampling structures and the BIG GRAPH

Sampling Algorithms $\equiv$ Random walk on BIG GRAPH. Small degree.

Vertices	Neighbors	Degree (ish)
Grid points in convex body.	Change one dimension	$2d$
Spanning Trees.	Change two edges.	$\leq \|V\|^2$ neighbors per node
Spin States.		

Sampling structures and the BIG GRAPH

Sampling Algorithms $\equiv$ Random walk on BIG GRAPH. Small degree.

Vertices	Neighbors	Degree (ish)
Grid points in convex body.	Change one dimension	$2d$
Spanning Trees.	Change two edges.	$\leq \|V\|^2$ neighbors per node
Spin States.	Change one spin	

Sampling structures and the BIG GRAPH

Sampling Algorithms $\equiv$ Random walk on BIG GRAPH. Small degree.

Vertices	Neighbors	Degree (ish)
Grid points in convex body.	Change one dimension	$2d$
Spanning Trees.	Change two edges.	$\leq \|V\|^2$ neighbors per node
Spin States.	Change one spin	$O(n)$ neighbors.

Analyzing random walks on graph.

Start at vertex, go to random neighbor.

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
  For d-regular graph: eventually uniform.

Analyzing random walks on graph.

Start at vertex, go to random neighbor.

For d-regular graph: eventually uniform.

if not bipartite.

Analyzing random walks on graph.

Start at vertex, go to random neighbor.

For *d*-regular graph: eventually uniform.

if not bipartite. Odd

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
  For d-regular graph: eventually uniform.
    if not bipartite. Odd /even step!

Analyzing random walks on graph.

Start at vertex, go to random neighbor.

For *d*-regular graph: eventually uniform.

if not bipartite. Odd /even step!

How to analyse?

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
For d-regular graph: eventually uniform.
if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
  For d-regular graph: eventually uniform.
    if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
  M - normalized adjacency matrix.

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
  For d-regular graph: eventually uniform.
    if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
  M - normalized adjacency matrix.
  Symmetric, $\sum_j M[i,j] = 1$.

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
For d-regular graph: eventually uniform.
if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
M - normalized adjacency matrix.
Symmetric, $\sum_j M[i,j] = 1$.
$M[i,j]$- probability of going to j from i.

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
For d-regular graph: eventually uniform.
if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
M - normalized adjacency matrix.
Symmetric, $\sum_j M[i,j] = 1$.
$M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
  For d-regular graph: eventually uniform.
    if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
  M - normalized adjacency matrix.
  Symmetric, $\sum_j M[i,j] = 1$.
  $M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.
  $v_{t+1} = Mv_t$

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
For d-regular graph: eventually uniform.
if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
M - normalized adjacency matrix.
Symmetric, $\sum_j M[i,j] = 1$.
$M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.
$v_{t+1} = Mv_t$ Each node is average over neighbors.

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
For d-regular graph: eventually uniform.
if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
M - normalized adjacency matrix.
Symmetric, $\sum_j M[i,j] = 1$.
$M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.
$v_{t+1} = Mv_t$ Each node is average over neighbors.

Evolution? Random walk

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
 For d-regular graph: eventually uniform.
  if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
 M - normalized adjacency matrix.
 Symmetric, $\sum_j M[i,j] = 1$.
 $M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.
 $v_{t+1} = Mv_t$ Each node is average over neighbors.

Evolution? Random walk starts at 1,

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
  For d-regular graph: eventually uniform.
    if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
  M - normalized adjacency matrix.
  Symmetric, $\sum_j M[i,j] = 1$.
  $M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.
  $v_{t+1} = Mv_t$ Each node is average over neighbors.

Evolution? Random walk starts at 1, distribution $e_1 = [1, 0, \ldots, 0]$.

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
For d-regular graph: eventually uniform.
if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
M - normalized adjacency matrix.
Symmetric, $\sum_j M[i,j] = 1$.
$M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.
$v_{t+1} = Mv_t$ Each node is average over neighbors.

Evolution? Random walk starts at 1, distribution $e_1 = [1, 0, \ldots, 0]$.
$M^t v_1 = \frac{1}{N} v_1 + \sum_{i>1} \lambda_i^t \alpha_i v_i$.

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
For d-regular graph: eventually uniform.
if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
M - normalized adjacency matrix.
Symmetric, $\sum_j M[i,j] = 1$.
$M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.
$v_{t+1} = Mv_t$ Each node is average over neighbors.

Evolution? Random walk starts at 1, distribution $e_1 = [1,0,\ldots,0]$.
$M^t v_1 = \frac{1}{N} v_1 + \sum_{i>1} \lambda_i^t \alpha_i v_i$.
$v_1 = [\frac{1}{N},\ldots,\frac{1}{N}]$

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
For d-regular graph: eventually uniform.
if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
M - normalized adjacency matrix.
Symmetric, $\sum_j M[i,j] = 1$.
$M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.
$v_{t+1} = Mv_t$ Each node is average over neighbors.

Evolution? Random walk starts at 1, distribution $e_1 = [1, 0, \ldots, 0]$.
$M^t v_1 = \frac{1}{N} v_1 + \sum_{i>1} \lambda_i^t \alpha_i v_i$.
$v_1 = [\frac{1}{N}, \ldots, \frac{1}{N}]$ $\rightarrow$Uniform distribution.

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
For d-regular graph: eventually uniform.
if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
M - normalized adjacency matrix.
Symmetric, $\sum_j M[i,j] = 1$.
$M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.
$v_{t+1} = Mv_t$ Each node is average over neighbors.

Evolution? Random walk starts at 1, distribution $e_1 = [1, 0, \ldots, 0]$.
$M^t v_1 = \frac{1}{N} v_1 + \sum_{i>1} \lambda_i^t \alpha_i v_i$.
$v_1 = [\frac{1}{N}, \ldots, \frac{1}{N}] \rightarrow$Uniform distribution.

Doh!

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
For d-regular graph: eventually uniform.
if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
M - normalized adjacency matrix.
Symmetric, $\sum_j M[i,j] = 1$.
$M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.
$v_{t+1} = Mv_t$ Each node is average over neighbors.

Evolution? Random walk starts at 1, distribution $e_1 = [1, 0, \ldots, 0]$.
$M^t v_1 = \frac{1}{N} v_1 + \sum_{i>1} \lambda_i^t \alpha_i v_i$.
$v_1 = [\frac{1}{N}, \ldots, \frac{1}{N}]$ $\rightarrow$Uniform distribution.

Doh! What if bipartite?

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
For d-regular graph: eventually uniform.
if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
M - normalized adjacency matrix.
Symmetric, $\sum_j M[i,j] = 1$.
$M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.
$v_{t+1} = Mv_t$ Each node is average over neighbors.

Evolution? Random walk starts at 1, distribution $e_1 = [1, 0, \ldots, 0]$.
$M^t v_1 = \frac{1}{N} v_1 + \sum_{i>1} \lambda_i^t \alpha_i v_i$.
$v_1 = [\frac{1}{N}, \ldots, \frac{1}{N}]$ $\rightarrow$Uniform distribution.

Doh! What if bipartite?
Negative eigenvalues of value -1:

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
For d-regular graph: eventually uniform.
if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
M - normalized adjacency matrix.
Symmetric, $\sum_j M[i,j] = 1$.
$M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.
$v_{t+1} = Mv_t$ Each node is average over neighbors.

Evolution? Random walk starts at 1, distribution $e_1 = [1, 0, \ldots, 0]$.
$M^t v_1 = \frac{1}{N} v_1 + \sum_{i>1} \lambda_i^t \alpha_i v_i$.
$v_1 = [\frac{1}{N}, \ldots, \frac{1}{N}]$ $\rightarrow$Uniform distribution.

Doh! What if bipartite?
Negative eigenvalues of value -1: $(+1, -1)$ on two sides.

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
For d-regular graph: eventually uniform.
if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
M - normalized adjacency matrix.
Symmetric, $\sum_j M[i,j] = 1$.
$M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.
$v_{t+1} = Mv_t$ Each node is average over neighbors.

Evolution? Random walk starts at 1, distribution $e_1 = [1, 0, \ldots, 0]$.
$M^t v_1 = \frac{1}{N} v_1 + \sum_{i>1} \lambda_i^t \alpha_i v_i$.
$v_1 = [\frac{1}{N}, \ldots, \frac{1}{N}]$ $\rightarrow$Uniform distribution.

Doh! What if bipartite?
Negative eigenvalues of value -1: $(+1, -1)$ on two sides.
Side question: Why the same size?

Analyzing random walks on graph.

Start at vertex, go to random neighbor.
For d-regular graph: eventually uniform.
if not bipartite. Odd /even step!

How to analyse?

Random Walk Matrix: M.
M - normalized adjacency matrix.
Symmetric, $\sum_j M[i,j] = 1$.
$M[i,j]$- probability of going to j from i.

Probability distribution at time t: v_t.
$v_{t+1} = Mv_t$ Each node is average over neighbors.

Evolution? Random walk starts at 1, distribution $e_1 = [1,0,\ldots,0]$.
$M^t v_1 = \frac{1}{N} v_1 + \sum_{i>1} \lambda_i^t \alpha_i v_i$.
$v_1 = [\frac{1}{N},\ldots,\frac{1}{N}] \rightarrow$Uniform distribution.

Doh! What if bipartite?
Negative eigenvalues of value -1: $(+1,-1)$ on two sides.
Side question: Why the same size? Assumed regular graph.

Fix-it-up chappie!

“Lazy” random walk:

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Fix-it-up chappie!

“Lazy” random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0, 1]$.

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0,1]$.

Spectral gap: $\frac{1-\lambda_2}{2} = \frac{\mu}{2}$.

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0,1]$.

Spectral gap: $\frac{1-\lambda_2}{2} = \frac{\mu}{2}$.

Uniform distribution: $\pi = [\frac{1}{N}, \ldots, \frac{1}{N}]$

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0, 1]$.

Spectral gap: $\frac{1-\lambda_2}{2} = \frac{\mu}{2}$.

Uniform distribution: $\pi = [\frac{1}{N}, \ldots, \frac{1}{N}]$

Distance to uniform:

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0,1]$.

Spectral gap: $\frac{1-\lambda_2}{2} = \frac{\mu}{2}$.

Uniform distribution: $\pi = [\frac{1}{N}, \ldots, \frac{1}{N}]$

Distance to uniform: $d_1(v_t, \pi) = \sum_i |(v_t)_i - \pi_i|$

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0,1]$.

Spectral gap: $\frac{1-\lambda_2}{2} = \frac{\mu}{2}$.

Uniform distribution: $\pi = [\frac{1}{N}, \ldots, \frac{1}{N}]$

Distance to uniform: $d_1(v_t, \pi) = \sum_i |(v_t)_i - \pi_i|$

"Rapidly mixing": $d_1(v_t, \pi) \leq \varepsilon$ in **poly**$(\log N, \log \frac{1}{\varepsilon})$ time.

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0,1]$.

Spectral gap: $\frac{1-\lambda_2}{2} = \frac{\mu}{2}$.

Uniform distribution: $\pi = [\frac{1}{N}, \ldots, \frac{1}{N}]$

Distance to uniform: $d_1(v_t, \pi) = \sum_i |(v_t)_i - \pi_i|$

"Rapidly mixing": $d_1(v_t, \pi) \le \varepsilon$ in **poly**$(\log N, \log \frac{1}{\varepsilon})$ time.

When is chain rapidly mixing?

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0,1]$.

Spectral gap: $\frac{1-\lambda_2}{2} = \frac{\mu}{2}$.

Uniform distribution: $\pi = [\frac{1}{N}, \ldots, \frac{1}{N}]$

Distance to uniform: $d_1(v_t, \pi) = \sum_i |(v_t)_i - \pi_i|$

"Rapidly mixing": $d_1(v_t, \pi) \le \varepsilon$ in **poly**$(\log N, \log \frac{1}{\varepsilon})$ time.

When is chain rapidly mixing?

Another measure:

Fix-it-up chappie!

“Lazy” random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0,1]$.

Spectral gap: $\frac{1-\lambda_2}{2} = \frac{\mu}{2}$.

Uniform distribution: $\pi = [\frac{1}{N}, \ldots, \frac{1}{N}]$

Distance to uniform: $d_1(v_t, \pi) = \sum_i |(v_t)_i - \pi_i|$

“Rapidly mixing”: $d_1(v_t, \pi) \leq \varepsilon$ in **poly**$(\log N, \log \frac{1}{\varepsilon})$ time.

When is chain rapidly mixing?

Another measure: $d_2(v_t, \pi) = \sum_i ((v_t)_i - \pi_i)^2$.

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0,1]$.

Spectral gap: $\frac{1-\lambda_2}{2} = \frac{\mu}{2}$.

Uniform distribution: $\pi = [\frac{1}{N}, \ldots, \frac{1}{N}]$

Distance to uniform: $d_1(v_t, \pi) = \sum_i |(v_t)_i - \pi_i|$

"Rapidly mixing": $d_1(v_t, \pi) \leq \varepsilon$ in **poly**$(\log N, \log \frac{1}{\varepsilon})$ time.

When is chain rapidly mixing?

Another measure: $d_2(v_t, \pi) = \sum_i ((v_t)_i - \pi_i)^2$.

Note: $d_1(v_t, \pi) \leq \sqrt{N} d_2(v_t, \pi)$

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0,1]$.

Spectral gap: $\frac{1-\lambda_2}{2} = \frac{\mu}{2}$.

Uniform distribution: $\pi = [\frac{1}{N}, \ldots, \frac{1}{N}]$

Distance to uniform: $d_1(v_t, \pi) = \sum_i |(v_t)_i - \pi_i|$

"Rapidly mixing": $d_1(v_t, \pi) \leq \varepsilon$ in **poly**$(\log N, \log \frac{1}{\varepsilon})$ time.

When is chain rapidly mixing?

Another measure: $d_2(v_t, \pi) = \sum_i ((v_t)_i - \pi_i)^2$.

Note: $d_1(v_t, \pi) \leq \sqrt{N} d_2(v_t, \pi)$

n – "size" of vertex,

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0,1]$.

Spectral gap: $\frac{1-\lambda_2}{2} = \frac{\mu}{2}$.

Uniform distribution: $\pi = [\frac{1}{N}, \ldots, \frac{1}{N}]$

Distance to uniform: $d_1(v_t, \pi) = \sum_i |(v_t)_i - \pi_i|$

"Rapidly mixing": $d_1(v_t, \pi) \leq \varepsilon$ in $\textbf{poly}(\log N, \log \frac{1}{\varepsilon})$ time.

When is chain rapidly mixing?

Another measure: $d_2(v_t, \pi) = \sum_i ((v_t)_i - \pi_i)^2$.

Note: $d_1(v_t, \pi) \leq \sqrt{N} d_2(v_t, \pi)$

n – "size" of vertex, $\mu \geq \frac{1}{p(n)}$ for poly $p(n)$,

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0,1]$.

Spectral gap: $\frac{1-\lambda_2}{2} = \frac{\mu}{2}$.

Uniform distribution: $\pi = [\frac{1}{N}, \ldots, \frac{1}{N}]$

Distance to uniform: $d_1(v_t, \pi) = \sum_i |(v_t)_i - \pi_i|$

"Rapidly mixing": $d_1(v_t, \pi) \leq \varepsilon$ in **poly**$(\log N, \log \frac{1}{\varepsilon})$ time.

When is chain rapidly mixing?

Another measure: $d_2(v_t, \pi) = \sum_i ((v_t)_i - \pi_i)^2$.

Note: $d_1(v_t, \pi) \leq \sqrt{N} d_2(v_t, \pi)$

n – "size" of vertex, $\mu \geq \frac{1}{p(n)}$ for poly $p(n)$, $t = O(p(n) \log N)$.

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0,1]$.

Spectral gap: $\frac{1-\lambda_2}{2} = \frac{\mu}{2}$.

Uniform distribution: $\pi = [\frac{1}{N}, \ldots, \frac{1}{N}]$

Distance to uniform: $d_1(v_t, \pi) = \sum_i |(v_t)_i - \pi_i|$

"Rapidly mixing": $d_1(v_t, \pi) \leq \varepsilon$ in **poly**$(\log N, \log \frac{1}{\varepsilon})$ time.

When is chain rapidly mixing?

Another measure: $d_2(v_t, \pi) = \sum_i ((v_t)_i - \pi_i)^2$.

Note: $d_1(v_t, \pi) \leq \sqrt{N} d_2(v_t, \pi)$

n – "size" of vertex, $\mu \geq \frac{1}{p(n)}$ for poly $p(n)$, $t = O(p(n) \log N)$.

$$d_2(v_t, \pi) = |A^t e_1 - \pi|^2 \leq \left(\frac{(1+\lambda_2)}{2}\right)^{2t} \leq (1 - \frac{1}{2p(n)})^{2t} \leq \frac{1}{\text{poly}(N)}$$

Fix-it-up chappie!

"Lazy" random walk: With probability $1/2$ stay at current vertex.

Evolution Matrix: $\frac{I+M}{2}$

Eigenvalues: $\frac{1+\lambda_i}{2}$

$\frac{1}{2}(I+M)v_i = \frac{1}{2}(v_i + \lambda_i v_i) = \frac{1+\lambda_i}{2} v_i$

Eigenvalues in interval $[0,1]$.

Spectral gap: $\frac{1-\lambda_2}{2} = \frac{\mu}{2}$.

Uniform distribution: $\pi = [\frac{1}{N}, \ldots, \frac{1}{N}]$

Distance to uniform: $d_1(v_t, \pi) = \sum_i |(v_t)_i - \pi_i|$

"Rapidly mixing": $d_1(v_t, \pi) \leq \varepsilon$ in $\mathbf{poly}(\log N, \log \frac{1}{\varepsilon})$ time.

When is chain rapidly mixing?

Another measure: $d_2(v_t, \pi) = \sum_i ((v_t)_i - \pi_i)^2$.

Note: $d_1(v_t, \pi) \leq \sqrt{N} d_2(v_t, \pi)$

n – "size" of vertex, $\mu \geq \frac{1}{p(n)}$ for poly $p(n)$, $t = O(p(n) \log N)$.

$$d_2(v_t, \pi) = |A^t e_1 - \pi|^2 \leq \left(\frac{(1+\lambda_2)}{2}\right)^{2t} \leq (1 - \frac{1}{2p(n)})^{2t} \leq \frac{1}{\text{poly}(N)}$$

Rapidly mixing with big ($\geq \frac{1}{p(n)}$) spectral gap.

Rapid mixing, volume, and surface area..

Recall volume of convex body.

Rapid mixing, volume, and surface area..

Recall volume of convex body.

Grid graph on grid points inside convex body.

Rapid mixing, volume, and surface area..

Recall volume of convex body.
  Grid graph on grid points inside convex body.

Recall Cheeger:

Rapid mixing, volume, and surface area..

Recall volume of convex body.
  Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \leq h(G) \leq \sqrt{2\mu}$.

Rapid mixing, volume, and surface area..

Recall volume of convex body.
  Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \leq h(G) \leq \sqrt{2\mu}$.

Lower bound expansion

Rapid mixing, volume, and surface area..

Recall volume of convex body.
  Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \leq h(G) \leq \sqrt{2\mu}$.

Lower bound expansion $\rightarrow$ lower bounds on spectral gap μ

Rapid mixing, volume, and surface area..

Recall volume of convex body.
  Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \leq h(G) \leq \sqrt{2\mu}$.

Lower bound expansion $\rightarrow$ lower bounds on spectral gap μ
  $\rightarrow$ Upper bound mixing time.

Rapid mixing, volume, and surface area..

Recall volume of convex body.
  Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \leq h(G) \leq \sqrt{2\mu}$.

Lower bound expansion $\rightarrow$ lower bounds on spectral gap μ
  $\rightarrow$ Upper bound mixing time.

$h(G) \approx \frac{\text{Surface Area}}{\text{Volume}}$

Rapid mixing, volume, and surface area..

Recall volume of convex body.
Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \leq h(G) \leq \sqrt{2\mu}$.

Lower bound expansion $\rightarrow$ lower bounds on spectral gap μ
$\rightarrow$ Upper bound mixing time.

$h(G) \approx \frac{\text{Surface Area}}{\text{Volume}}$

Rapid mixing, volume, and surface area..

Recall volume of convex body.
  Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \leq h(G) \leq \sqrt{2\mu}$.

Lower bound expansion $\rightarrow$ lower bounds on spectral gap μ
  $\rightarrow$ Upper bound mixing time.

$h(G) \approx \frac{\text{Surface Area}}{\text{Volume}}$

Isoperimetric inequality.

Rapid mixing, volume, and surface area..

Recall volume of convex body.
Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \leq h(G) \leq \sqrt{2\mu}$.

Lower bound expansion $\rightarrow$ lower bounds on spectral gap μ
$\rightarrow$ Upper bound mixing time.

$h(G) \approx \frac{\text{Surface Area}}{\text{Volume}}$

Isoperimetric inequality.

$\text{Vol}_{n-1}(S, \overline{S}) \geq \frac{\min(\text{Vol}(S), \text{Vol}(\overline{S}))}{diam(P)}$

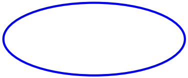
Rapid mixing, volume, and surface area..

Recall volume of convex body.
Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \le h(G) \le \sqrt{2\mu}$.

Lower bound expansion $\rightarrow$ lower bounds on spectral gap μ
$\rightarrow$ Upper bound mixing time.

$h(G) \approx \frac{\text{Surface Area}}{\text{Volume}}$



Isoperimetric inequality.

$\text{Vol}_{n-1}(S, \overline{S}) \ge \frac{\min(\text{Vol}(S), \text{Vol}(\overline{S}))}{diam(P)}$

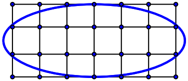
Rapid mixing, volume, and surface area..

Recall volume of convex body.
Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \le h(G) \le \sqrt{2\mu}$.

Lower bound expansion $\rightarrow$ lower bounds on spectral gap μ
$\rightarrow$ Upper bound mixing time.

$h(G) \approx \frac{\text{Surface Area}}{\text{Volume}}$



Isoperimetric inequality.

$\text{Vol}_{n-1}(S, \overline{S}) \ge \frac{\min(\text{Vol}(S), \text{Vol}(\overline{S}))}{diam(P)}$

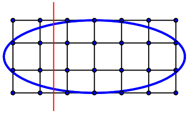
Rapid mixing, volume, and surface area..

Recall volume of convex body.
Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \leq h(G) \leq \sqrt{2\mu}$.

Lower bound expansion $\rightarrow$ lower bounds on spectral gap μ
$\rightarrow$ Upper bound mixing time.

$h(G) \approx \frac{\text{Surface Area}}{\text{Volume}}$



Isoperimetric inequality.

$\text{Vol}_{n-1}(S, \overline{S}) \geq \frac{\min(\text{Vol}(S), \text{Vol}(\overline{S}))}{diam(P)}$

Edges $\propto$ surface area,

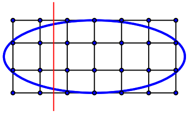
Rapid mixing, volume, and surface area..

Recall volume of convex body.
Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \leq h(G) \leq \sqrt{2\mu}$.

Lower bound expansion $\rightarrow$ lower bounds on spectral gap μ
$\rightarrow$ Upper bound mixing time.

$h(G) \approx \frac{\text{Surface Area}}{\text{Volume}}$



Isoperimetric inequality.

$\text{Vol}_{n-1}(S, \overline{S}) \geq \frac{\min(\text{Vol}(S), \text{Vol}(\overline{S}))}{diam(P)}$

Edges $\propto$ surface area, Assume $Diam(P) \leq p'(n)$

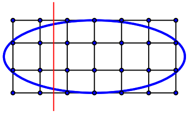
Rapid mixing, volume, and surface area..

Recall volume of convex body.
Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \leq h(G) \leq \sqrt{2\mu}$.

Lower bound expansion $\rightarrow$ lower bounds on spectral gap μ
$\rightarrow$ Upper bound mixing time.

$h(G) \approx \frac{\text{Surface Area}}{\text{Volume}}$



Isoperimetric inequality.

$\text{Vol}_{n-1}(S, \overline{S}) \geq \frac{\min(\text{Vol}(S), \text{Vol}(\overline{S}))}{diam(P)}$

Edges $\propto$ surface area, Assume $Diam(P) \leq p'(n)$
$\rightarrow h(G) \geq 1/p'(n)$

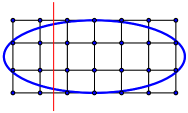
Rapid mixing, volume, and surface area..

Recall volume of convex body.
Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \leq h(G) \leq \sqrt{2\mu}$.

Lower bound expansion $\rightarrow$ lower bounds on spectral gap μ
$\rightarrow$ Upper bound mixing time.

$h(G) \approx \frac{\text{Surface Area}}{\text{Volume}}$



Isoperimetric inequality.

$\text{Vol}_{n-1}(S, \overline{S}) \geq \frac{\min(\text{Vol}(S), \text{Vol}(\overline{S}))}{diam(P)}$

Edges $\propto$ surface area, Assume $Diam(P) \leq p'(n)$
$\rightarrow h(G) \geq 1/p'(n)$
$\rightarrow \mu > 1/2p'(n)^2$

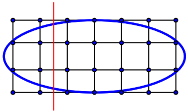
Rapid mixing, volume, and surface area..

Recall volume of convex body.
Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \leq h(G) \leq \sqrt{2\mu}$.

Lower bound expansion $\rightarrow$ lower bounds on spectral gap μ
$\rightarrow$ Upper bound mixing time.

$h(G) \approx \frac{\text{Surface Area}}{\text{Volume}}$



Isoperimetric inequality.

$\text{Vol}_{n-1}(S, \overline{S}) \geq \frac{\min(\text{Vol}(S), \text{Vol}(\overline{S}))}{diam(P)}$

Edges $\propto$ surface area, Assume $Diam(P) \leq p'(n)$
$\rightarrow h(G) \geq 1/p'(n)$
$\rightarrow \mu > 1/2p'(n)^2$
$\rightarrow O(p'(n)^2 \log N)$ convergence for Markov chain on BIG GRAPH.

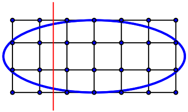
Rapid mixing, volume, and surface area..

Recall volume of convex body.
Grid graph on grid points inside convex body.

Recall Cheeger: $\frac{\mu}{2} \leq h(G) \leq \sqrt{2\mu}$.

Lower bound expansion $\rightarrow$ lower bounds on spectral gap μ
$\rightarrow$ Upper bound mixing time.

$h(G) \approx \frac{\text{Surface Area}}{\text{Volume}}$



Isoperimetric inequality.

$$Vol_{n-1}(S, \overline{S}) \geq \frac{\min(Vol(S), Vol(\overline{S}))}{diam(P)}$$

Edges $\propto$ surface area, Assume $Diam(P) \leq p'(n)$
$\rightarrow h(G) \geq 1/p'(n)$
$\rightarrow \mu > 1/2p'(n)^2$
$\rightarrow O(p'(n)^2 \log N)$ convergence for Markov chain on BIG GRAPH.
$\rightarrow$ Rapidly mixing chain:

Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

Sample from uniform distribution over total orders.

Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

Sample from uniform distribution over total orders.

Start at an ordering.

Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

Sample from uniform distribution over total orders.

Start at an ordering.
Swap random pair

Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

Sample from uniform distribution over total orders.

Start at an ordering.
Swap random pair and go if consistent with partial order.

Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

Sample from uniform distribution over total orders.

Start at an ordering.
Swap random pair and go if consistent with partial order.

Rapidly mixing chain?

Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

Sample from uniform distribution over total orders.

Start at an ordering.
Swap random pair and go if consistent with partial order.

Rapidly mixing chain?

Map into d-dimensional unit cube.

Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

Sample from uniform distribution over total orders.

Start at an ordering.
Swap random pair and go if consistent with partial order.

Rapidly mixing chain?

Map into d-dimensional unit cube.

$x_i < x_j$ corresponds to halfspace (one side of hyperplane) of cube.

Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

Sample from uniform distribution over total orders.

Start at an ordering.
Swap random pair and go if consistent with partial order.

Rapidly mixing chain?

Map into d-dimensional unit cube.

$x_i < x_j$ corresponds to halfspace (one side of hyperplane) of cube.
"dimension i = dimension j"

Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

Sample from uniform distribution over total orders.

Start at an ordering.
Swap random pair and go if consistent with partial order.

Rapidly mixing chain?

Map into d-dimensional unit cube.

$x_i < x_j$ corresponds to halfspace (one side of hyperplane) of cube.
"dimension i = dimension j"
total order is intersection of n halfspaces.

Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

Sample from uniform distribution over total orders.

Start at an ordering.
Swap random pair and go if consistent with partial order.

Rapidly mixing chain?

Map into d-dimensional unit cube.

$x_i < x_j$ corresponds to halfspace (one side of hyperplane) of cube.
"dimension i = dimension j"
total order is intersection of n halfspaces.
each of volume: $\frac{1}{n!}$.

Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

Sample from uniform distribution over total orders.

Start at an ordering.
Swap random pair and go if consistent with partial order.

Rapidly mixing chain?

Map into d-dimensional unit cube.

$x_i < x_j$ corresponds to halfspace (one side of hyperplane) of cube.
"dimension i = dimension j"
total order is intersection of n halfspaces.
each of volume: $\frac{1}{n!}$.
since each total order is disjoint

Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

Sample from uniform distribution over total orders.

Start at an ordering.
Swap random pair and go if consistent with partial order.

Rapidly mixing chain?

Map into d-dimensional unit cube.

$x_i < x_j$ corresponds to halfspace (one side of hyperplane) of cube.
"dimension i = dimension j"
total order is intersection of n halfspaces.
each of volume: $\frac{1}{n!}$.
since each total order is disjoint
and together cover cube.

Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

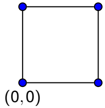
Sample from uniform distribution over total orders.

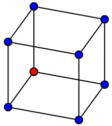
Start at an ordering.
Swap random pair and go if consistent with partial order.

Rapidly mixing chain?

Map into d-dimensional unit cube.

$x_i < x_j$ corresponds to halfspace (one side of hyperplane) of cube.
"dimension i = dimension j"
total order is intersection of n halfspaces.
each of volume: $\frac{1}{n!}$.
since each total order is disjoint
and together cover cube.





Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

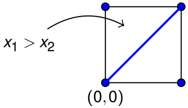
Sample from uniform distribution over total orders.

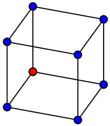
Start at an ordering.
Swap random pair and go if consistent with partial order.

Rapidly mixing chain?

Map into d-dimensional unit cube.

$x_i < x_j$ corresponds to halfspace (one side of hyperplane) of cube.
"dimension i = dimension j"
total order is intersection of n halfspaces.
each of volume: $\frac{1}{n!}$.
since each total order is disjoint
and together cover cube.





Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

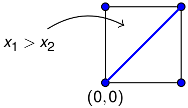
Sample from uniform distribution over total orders.

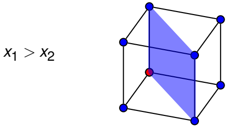
Start at an ordering.
Swap random pair and go if consistent with partial order.

Rapidly mixing chain?

Map into d-dimensional unit cube.

$x_i < x_j$ corresponds to halfspace (one side of hyperplane) of cube.
"dimension i = dimension j"
total order is intersection of n halfspaces.
each of volume: $\frac{1}{n!}$.
since each total order is disjoint
and together cover cube.





Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

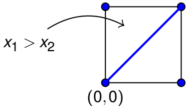
Sample from uniform distribution over total orders.

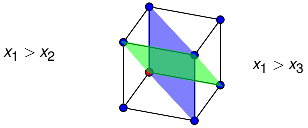
Start at an ordering.
Swap random pair and go if consistent with partial order.

Rapidly mixing chain?

Map into d-dimensional unit cube.

$x_i < x_j$ corresponds to halfspace (one side of hyperplane) of cube.
"dimension i = dimension j"
total order is intersection of n halfspaces.
each of volume: $\frac{1}{n!}$.
since each total order is disjoint
and together cover cube.





Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

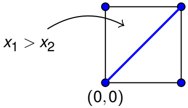
Sample from uniform distribution over total orders.

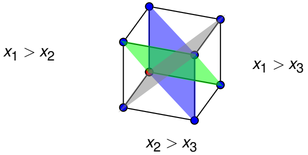
Start at an ordering.
Swap random pair and go if consistent with partial order.

Rapidly mixing chain?

Map into d-dimensional unit cube.

$x_i < x_j$ corresponds to halfspace (one side of hyperplane) of cube.
"dimension i = dimension j"
total order is intersection of n halfspaces.
each of volume: $\frac{1}{n!}$.
since each total order is disjoint
and together cover cube.





Khachiyan's algorithm for counting partial orders.

Given partial order on $x_1, \ldots, x_n$.

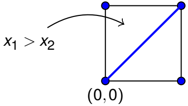
Sample from uniform distribution over total orders.

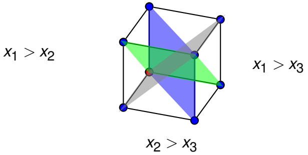
Start at an ordering.
Swap random pair and go if consistent with partial order.

Rapidly mixing chain?

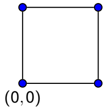
Map into d-dimensional unit cube.

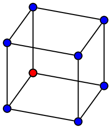
$x_i < x_j$ corresponds to halfspace (one side of hyperplane) of cube.
"dimension i = dimension j"
total order is intersection of n halfspaces.
each of volume: $\frac{1}{n!}$.
since each total order is disjoint
and together cover cube.



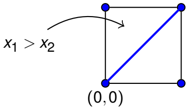


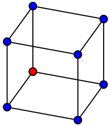
Poly time.



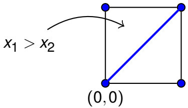


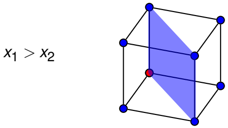
Poly time.



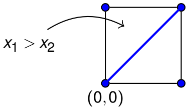


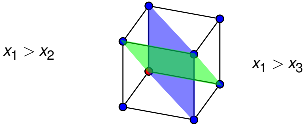
Poly time.



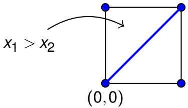


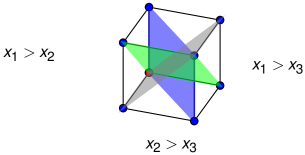
Poly time.



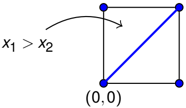


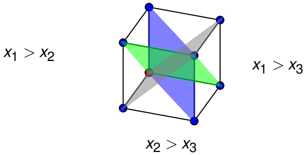
Poly time.





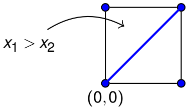
Poly time.

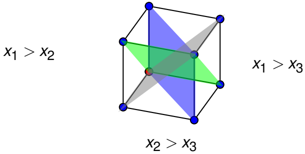




Each order takes $\frac{1}{n!}$ volume.

Poly time.

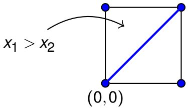


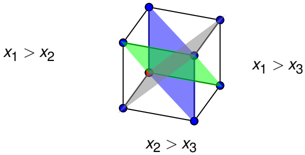


Each order takes $\frac{1}{n!}$ volume.

Number of orders $\equiv$ volume of intersection of partial order relations.

Poly time.



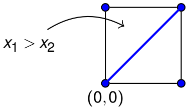


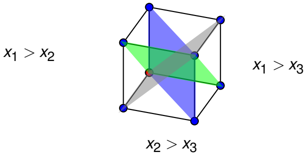
Each order takes $\frac{1}{n!}$ volume.

Number of orders $\equiv$ volume of intersection of partial order relations.

Diameter: $O(\sqrt{n})$

Poly time.





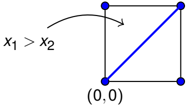
Each order takes $\frac{1}{n!}$ volume.

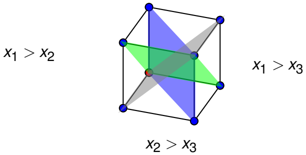
Number of orders $\equiv$ volume of intersection of partial order relations.

Diameter: $O(\sqrt{n})$

Isoperimetry:

Poly time.





Each order takes $\frac{1}{n!}$ volume.

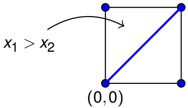
Number of orders $\equiv$ volume of intersection of partial order relations.

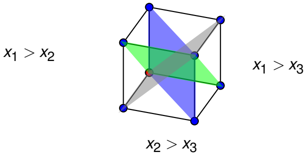
Diameter: $O(\sqrt{n})$

Isoperimetry:

$Vol_n(S) = \frac{|S|}{n!}$.

Poly time.





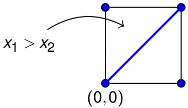
Each order takes $\frac{1}{n!}$ volume.

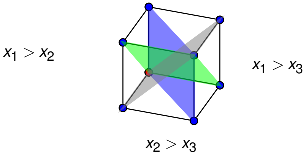
Number of orders $\equiv$ volume of intersection of partial order relations.

Diameter: $O(\sqrt{n})$

Isoperimetry:
$Vol_n(S) = \frac{|S|}{n!}$. and $Vol_{n-1}(S, \overline{S}) \geq \frac{Vol_n S}{diam(P)}$.

Poly time.





Each order takes $\frac{1}{n!}$ volume.

Number of orders $\equiv$ volume of intersection of partial order relations.

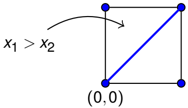
Diameter: $O(\sqrt{n})$

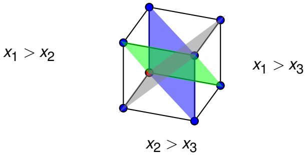
Isoperimetry:

$Vol_n(S) = \frac{|S|}{n!}$. and $Vol_{n-1}(S,\overline{S}) \geq \frac{Vol_n S}{diam(P)}$.

$$Vol_{n-1}(S,\overline{S}) = \frac{E(S,\overline{S})}{(n-1)!}$$

Poly time.





Each order takes $\frac{1}{n!}$ volume.

Number of orders $\equiv$ volume of intersection of partial order relations.

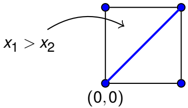
Diameter: $O(\sqrt{n})$

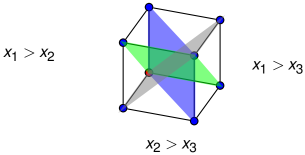
Isoperimetry:

$Vol_n(S) = \frac{|S|}{n!}$. and $Vol_{n-1}(S, \overline{S}) \geq \frac{Vol_n S}{diam(P)}$.

$$Vol_{n-1}(S, \overline{S}) = \frac{E(S, \overline{S})}{(n-1)!} \geq \frac{|S|}{n!\sqrt{n}}$$

Poly time.





Each order takes $\frac{1}{n!}$ volume.

Number of orders $\equiv$ volume of intersection of partial order relations.

Diameter: $O(\sqrt{n})$

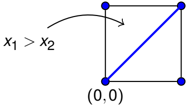
Isoperimetry:

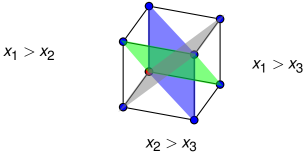
$Vol_n(S) = \frac{|S|}{n!}$. and $Vol_{n-1}(S, \overline{S}) \geq \frac{Vol_n S}{diam(P)}$.

$$Vol_{n-1}(S, \overline{S}) = \frac{E(S, \overline{S})}{(n-1)!} \geq \frac{|S|}{n!\sqrt{n}}$$

Edge Expansion: the degree d is $O(n^2)$,

Poly time.





Each order takes $\frac{1}{n!}$ volume.

Number of orders $\equiv$ volume of intersection of partial order relations.

Diameter: $O(\sqrt{n})$

Isoperimetry:

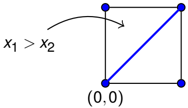
$Vol_n(S) = \frac{|S|}{n!}$. and $Vol_{n-1}(S, \overline{S}) \geq \frac{Vol_n S}{diam(P)}$.

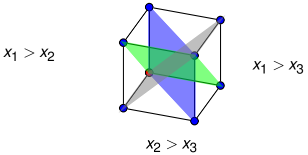
$$Vol_{n-1}(S, \overline{S}) = \frac{E(S, \overline{S})}{(n-1)!} \geq \frac{|S|}{n!\sqrt{n}}$$

Edge Expansion: the degree d is $O(n^2)$,

$h(S) = \frac{|E(S, \overline{S})|}{d|S|} \geq \frac{1}{n^{7/2}}$

Poly time.





Each order takes $\frac{1}{n!}$ volume.

Number of orders $\equiv$ volume of intersection of partial order relations.

Diameter: $O(\sqrt{n})$

Isoperimetry:

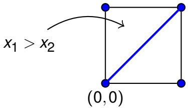
$Vol_n(S) = \frac{|S|}{n!}$. and $Vol_{n-1}(S, \overline{S}) \geq \frac{Vol_n S}{diam(P)}$.

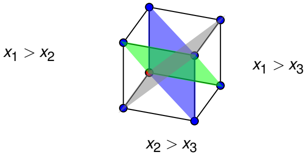
$$Vol_{n-1}(S, \overline{S}) = \frac{E(S, \overline{S})}{(n-1)!} \geq \frac{|S|}{n!\sqrt{n}}$$

Edge Expansion: the degree d is $O(n^2)$,

$h(S) = \frac{|E(S,\overline{S})|}{d|S|} \geq \frac{1}{n^{7/2}}$ Mixes in time $O(n^7 \log N)$

Poly time.





Each order takes $\frac{1}{n!}$ volume.

Number of orders $\equiv$ volume of intersection of partial order relations.

Diameter: $O(\sqrt{n})$

Isoperimetry:

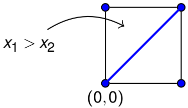
$Vol_n(S) = \frac{|S|}{n!}$. and $Vol_{n-1}(S, \overline{S}) \geq \frac{Vol_n S}{diam(P)}$.

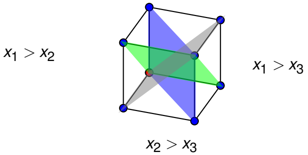
$$Vol_{n-1}(S, \overline{S}) = \frac{E(S,\overline{S})}{(n-1)!} \geq \frac{|S|}{n!\sqrt{n}}$$

Edge Expansion: the degree d is $O(n^2)$,

$h(S) = \frac{|E(S,\overline{S})|}{d|S|} \geq \frac{1}{n^{7/2}}$ Mixes in time $O(n^7 \log N)$ = $O(n^8 \log n)$.

Poly time.





Each order takes $\frac{1}{n!}$ volume.

Number of orders $\equiv$ volume of intersection of partial order relations.

Diameter: $O(\sqrt{n})$

Isoperimetry:

$Vol_n(S) = \frac{|S|}{n!}$. and $Vol_{n-1}(S, \overline{S}) \geq \frac{Vol_n S}{diam(P)}$.

$$Vol_{n-1}(S, \overline{S}) = \frac{E(S, \overline{S})}{(n-1)!} \geq \frac{|S|}{n!\sqrt{n}}$$

Edge Expansion: the degree d is $O(n^2)$,

$h(S) = \frac{|E(S,\overline{S})|}{d|S|} \geq \frac{1}{n^{7/2}}$ Mixes in time $O(n^7 \log N)$ = $O(n^8 \log n)$.

Do the polynomial dance!!!

Miller-Rabin.

Pick a random a, check if $a^{N-1} = 1 \pmod{N}$.

Miller-Rabin.

Pick a random a, check if $a^{N-1} = 1 \pmod{N}$.

If N not prime and any a fails test, half the a's fail test.

Miller-Rabin.

Pick a random a, check if $a^{N-1} = 1 \pmod{N}$.

If N not prime and any a fails test, half the a's fail test.

Repeat k times.

Miller-Rabin.

Pick a random a, check if $a^{N-1} = 1 \pmod{N}$.

If N not prime and any a fails test, half the a's fail test.

Repeat k times.

n possibilities, $\log n$ bits, half the possibilities are good.

Miller-Rabin.

Pick a random a, check if $a^{N-1} = 1 \pmod{N}$.

If N not prime and any a fails test, half the a's fail test.

Repeat k times.

n possibilities, $\log n$ bits, half the possibilities are good.

Total: $O(k \log n)$ random bits.

Miller-Rabin.

Pick a random a, check if $a^{N-1} = 1 \pmod{N}$.

If N not prime and any a fails test, half the a's fail test.

Repeat k times.

n possibilities, $\log n$ bits, half the possibilities are good.

Total: $O(k \log n)$ random bits. Failure probability is $1/2^k$.

Miller-Rabin.

Pick a random a, check if $a^{N-1} = 1 \pmod{N}$.

If N not prime and any a fails test, half the a's fail test.

Repeat k times.

n possibilities, $\log n$ bits, half the possibilities are good.

Total: $O(k \log n)$ random bits. Failure probability is $1/2^k$.

Another view: n-vertex degree d graph with $\lambda_1 - \lambda_2 \geq \Omega(\sqrt{d})$.

Miller-Rabin.

Pick a random a, check if $a^{N-1} = 1 \pmod{N}$.

If N not prime and any a fails test, half the a's fail test.

Repeat k times.

n possibilities, $\log n$ bits, half the possibilities are good.

Total: $O(k \log n)$ random bits. Failure probability is $1/2^k$.

Another view: n-vertex degree d graph with $\lambda_1 - \lambda_2 \geq \Omega(\sqrt{d})$.

Half the vertices correspond to good a's.

Miller-Rabin.

Pick a random a, check if $a^{N-1} = 1 \pmod{N}$.

If N not prime and any a fails test, half the a's fail test.

Repeat k times.

n possibilities, $\log n$ bits, half the possibilities are good.

Total: $O(k \log n)$ random bits. Failure probability is $1/2^k$.

Another view: n-vertex degree d graph with $\lambda_1 - \lambda_2 \geq \Omega(\sqrt{d})$.

Half the vertices correspond to good a's.

Choose random vertex, and do random walk of length ck.

Miller-Rabin.

Pick a random a, check if $a^{N-1} = 1 \pmod{N}$.

If N not prime and any a fails test, half the a's fail test.

Repeat k times.

n possibilities, $\log n$ bits, half the possibilities are good.

Total: $O(k \log n)$ random bits. Failure probability is $1/2^k$.

Another view: n-vertex degree d graph with $\lambda_1 - \lambda_2 \geq \Omega(\sqrt{d})$.

Half the vertices correspond to good a's.

Choose random vertex, and do random walk of length ck.
$\ell = \log n + ck$ random bits.

Miller-Rabin.

Pick a random a, check if $a^{N-1} = 1 \pmod{N}$.

If N not prime and any a fails test, half the a's fail test.

Repeat k times.

n possibilities, $\log n$ bits, half the possibilities are good.

Total: $O(k \log n)$ random bits. Failure probability is $1/2^k$.

Another view: n-vertex degree d graph with $\lambda_1 - \lambda_2 \geq \Omega(\sqrt{d})$.

Half the vertices correspond to good a's.

Choose random vertex, and do random walk of length ck.
$\ell = \log n + ck$ random bits.

Given a random walk of length ℓ in an expander graph (N, d, λ), what is the probability that you stay in a bad set B, with $|B|/n \leq \beta = \frac{1}{2}$, for all the steps?

Miller-Rabin.

Pick a random a, check if $a^{N-1} = 1 \pmod{N}$.

If N not prime and any a fails test, half the a's fail test.

Repeat k times.

n possibilities, $\log n$ bits, half the possibilities are good.

Total: $O(k \log n)$ random bits. Failure probability is $1/2^k$.

Another view: n-vertex degree d graph with $\lambda_1 - \lambda_2 \geq \Omega(\sqrt{d})$.

Half the vertices correspond to good a's.

Choose random vertex, and do random walk of length ck.
$\ell = \log n + ck$ random bits.

Given a random walk of length ℓ in an expander graph (N, d, λ), what is the probability that you stay in a bad set B, with $|B|/n \leq \beta = \frac{1}{2}$, for all the steps?

$Pr[\text{ stay in } B] \leq ((1-\lambda)\sqrt{\beta} + \lambda)^{\ell}$

Proof: set up walk.

Claim: $Pr[\text{ stay in } B] \leq ((1-\lambda)\sqrt{\beta}+\lambda)^{\ell}$

Proof: set up walk.

Claim: $Pr[\text{ stay in } B] \leq ((1-\lambda)\sqrt{\beta}+\lambda)^{\ell}$

B_i - event in B at step i.

Proof: set up walk.

Claim: $Pr[\text{ stay in } B] \leq ((1-\lambda)\sqrt{\beta}+\lambda)^{\ell}$

B_i - event in B at step i.

$\hat{B}$ is diagonal matrix with 1's corresponding to $i \in B$.

Proof: set up walk.

Claim: $Pr[\text{ stay in } B] \leq ((1-\lambda)\sqrt{\beta}+\lambda)^{\ell}$

B_i - event in B at step i.

$\hat{B}$ is diagonal matrix with 1's corresponding to $i \in B$.

Consider random walk that truncates when it hits $v \notin B$.

Proof: set up walk.

Claim: $Pr[\text{ stay in } B] \leq ((1-\lambda)\sqrt{\beta}+\lambda)^{\ell}$

B_i - event in B at step i.

$\hat{B}$ is diagonal matrix with 1's corresponding to $i \in B$.

Consider random walk that truncates when it hits $v \notin B$.

Distribution over B at beginning: $\hat{B}\mathbf{1}$.

Proof: set up walk.

Claim: $Pr[\text{ stay in } B] \leq ((1-\lambda)\sqrt{\beta}+\lambda)^{\ell}$

B_i - event in B at step i.

$\hat{B}$ is diagonal matrix with 1's corresponding to $i \in B$.

Consider random walk that truncates when it hits $v \notin B$.

Distribution over B at beginning: $\hat{B}\mathbf{1}$. $\mathbf{1} = (\mathbf{1}/\mathbf{N}, \ldots, \mathbf{1}/\mathbf{N})$

Proof: set up walk.

Claim: $Pr[\text{ stay in } B] \leq ((1-\lambda)\sqrt{\beta}+\lambda)^{\ell}$

B_i - event in B at step i.

$\hat{B}$ is diagonal matrix with 1's corresponding to $i \in B$.

Consider random walk that truncates when it hits $v \notin B$.

Distribution over B at beginning: $\hat{B}\mathbf{1}$. $\mathbf{1} = (\mathbf{1/N}, \ldots, \mathbf{1/N})$
$1/N$ for each vertex in B.

Proof: set up walk.

Claim: $Pr[\text{ stay in } B] \leq ((1-\lambda)\sqrt{\beta}+\lambda)^{\ell}$

B_i - event in B at step i.

$\hat{B}$ is diagonal matrix with 1's corresponding to $i \in B$.

Consider random walk that truncates when it hits $v \notin B$.

Distribution over B at beginning: $\hat{B}\mathbf{1}$. $\mathbf{1} = (\mathbf{1/N}, \ldots, \mathbf{1/N})$
$1/N$ for each vertex in B.

Distribution over B at time 2,

Proof: set up walk.

Claim: $Pr[\text{ stay in } B] \leq ((1-\lambda)\sqrt{\beta}+\lambda)^{\ell}$

B_i - event in B at step i.

$\hat{B}$ is diagonal matrix with 1's corresponding to $i \in B$.

Consider random walk that truncates when it hits $v \notin B$.

Distribution over B at beginning: $\hat{B}\mathbf{1}$. $\mathbf{1} = (\mathbf{1/N}, \ldots, \mathbf{1/N})$
 $1/N$ for each vertex in B.

Distribution over B at time 2, $\hat{B}A\hat{B}\mathbf{1}$

Proof: set up walk.

Claim: $Pr[\text{ stay in } B] \leq ((1-\lambda)\sqrt{\beta}+\lambda)^{\ell}$

B_i - event in B at step i.

$\hat{B}$ is diagonal matrix with 1's corresponding to $i \in B$.

Consider random walk that truncates when it hits $v \notin B$.

Distribution over B at beginning: $\hat{B}\mathbf{1}$. $\mathbf{1} = (\mathbf{1/N}, \ldots, \mathbf{1/N})$
$1/N$ for each vertex in B.

Distribution over B at time 2, $\hat{B}A\hat{B}\mathbf{1}$

At time ℓ, $(\hat{B}A)^{\ell}\hat{B}\mathbf{1}$.

Proof: set up walk.

Claim: $Pr[\text{ stay in } B] \leq ((1-\lambda)\sqrt{\beta}+\lambda)^{\ell}$

B_i - event in B at step i.

$\hat{B}$ is diagonal matrix with 1's corresponding to $i \in B$.

Consider random walk that truncates when it hits $v \notin B$.

Distribution over B at beginning: $\hat{B}\mathbf{1}$. $\mathbf{1} = (\mathbf{1/N}, \ldots, \mathbf{1/N})$
$1/N$ for each vertex in B.

Distribution over B at time 2, $\hat{B}A\hat{B}\mathbf{1}$

At time ℓ, $(\hat{B}A)^{\ell}\hat{B}\mathbf{1}$.

Total probability in B: $\|(\hat{B}A)^{\ell}\hat{B}\mathbf{1}\|_{\mathbf{1}}$

Proof: set up walk.

Claim: $Pr[\text{ stay in } B] \leq ((1-\lambda)\sqrt{\beta}+\lambda)^{\ell}$

B_i - event in B at step i.

$\hat{B}$ is diagonal matrix with 1's corresponding to $i \in B$.

Consider random walk that truncates when it hits $v \notin B$.

Distribution over B at beginning: $\hat{B}\mathbf{1}$. $\mathbf{1} = (\mathbf{1}/\mathbf{N}, \ldots, \mathbf{1}/\mathbf{N})$
$1/N$ for each vertex in B.

Distribution over B at time 2, $\hat{B}A\hat{B}\mathbf{1}$

At time ℓ, $(\hat{B}A)^{\ell}\hat{B}\mathbf{1}$.

Total probability in B: $\|(\hat{B}A)^{\ell}\hat{B}\mathbf{1}\|_{\mathbf{1}}$

Will prove: $\|(\hat{B}A)^{\ell}\hat{B}\mathbf{1}\|_{\mathbf{2}} \leq \frac{((\mathbf{1}-\lambda)\sqrt{\beta}+\lambda)^{\ell}\sqrt{\beta}}{\sqrt{\mathbf{N}}}$

Proof: set up walk.

Claim: $Pr[$ stay in $B] \le ((1-\lambda)\sqrt{\beta}+\lambda)^{\ell}$

B_i - event in B at step i.

$\hat{B}$ is diagonal matrix with 1's corresponding to $i \in B$.

Consider random walk that truncates when it hits $v \notin B$.

Distribution over B at beginning: $\hat{B}\mathbf{1}$. $\mathbf{1} = (\mathbf{1/N}, \ldots, \mathbf{1/N})$
$1/N$ for each vertex in B.

Distribution over B at time 2, $\hat{B}A\hat{B}\mathbf{1}$

At time ℓ, $(\hat{B}A)^{\ell}\hat{B}\mathbf{1}$.

Total probability in B: $\|(\hat{B}A)^{\ell}\hat{B}\mathbf{1}\|_{\mathbf{1}}$

Will prove: $\|(\hat{B}A)^{\ell}\hat{B}\mathbf{1}\|_{\mathbf{2}} \le \frac{((\mathbf{1}-\lambda)\sqrt{\beta}+\lambda)^{\ell}\sqrt{\beta}}{\sqrt{\mathbf{N}}}$

plus $|x|_1 \le \sqrt{N}|x|_2$

Proof: set up walk.

Claim: $Pr[\text{ stay in } B] \leq ((1-\lambda)\sqrt{\beta}+\lambda)^{\ell}$

B_i - event in B at step i.

$\hat{B}$ is diagonal matrix with 1's corresponding to $i \in B$.

Consider random walk that truncates when it hits $v \notin B$.

Distribution over B at beginning: $\hat{B}\mathbf{1}$. $\mathbf{1} = (\mathbf{1}/\mathbf{N}, \ldots, \mathbf{1}/\mathbf{N})$
$1/N$ for each vertex in B.

Distribution over B at time 2, $\hat{B}A\hat{B}\mathbf{1}$

At time ℓ, $(\hat{B}A)^{\ell}\hat{B}\mathbf{1}$.

Total probability in B: $\|(\hat{B}A)^{\ell}\hat{B}\mathbf{1}\|_{\mathbf{1}}$

Will prove: $\|(\hat{B}A)^{\ell}\hat{B}\mathbf{1}\|_{\mathbf{2}} \leq \frac{((\mathbf{1}-\lambda)\sqrt{\beta}+\lambda)^{\ell}\sqrt{\beta}}{\sqrt{\mathbf{N}}}$

plus $|x|_1 \leq \sqrt{N}|x|_2 \implies$ Claim.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2 \|B\|_2$.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2 \|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2\|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2 \|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$A = (1-\lambda)J + \lambda C$.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2 \|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$$A = (1-\lambda)J + \lambda C.$$

where $\|Cv\|_2 \leq \|v\|_2$ for all v.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2\|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$A = (1-\lambda)J + \lambda C$.

where $\|Cv\|_2 \leq \|v\|_2$ for all v.

Proof:

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2\|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$A = (1-\lambda)J + \lambda C$.

where $\|Cv\|_2 \leq \|v\|_2$ for all v.

Proof:
$C = \frac{1}{\lambda}(A - (1-\lambda)J)$.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2 \|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$A = (1-\lambda)J + \lambda C$.

where $\|Cv\|_2 \leq \|v\|_2$ for all v.

Proof:
$C = \frac{1}{\lambda}(A - (1-\lambda)J)$.
Consider $v = u + w$, with $u = \alpha 1$ and $w \perp 1$.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2 \|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$A = (1-\lambda)J + \lambda C$.

where $\|Cv\|_2 \leq \|v\|_2$ for all v.

Proof:
$C = \frac{1}{\lambda}(A - (1-\lambda)J)$.
Consider $v = u + w$, with $u = \alpha 1$ and $w \perp 1$.
$Cu = \frac{1}{\lambda}(A - (1-\lambda)J)u = \frac{1}{\lambda}(1-(1-\lambda))u = u$

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2\|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$A = (1-\lambda)J + \lambda C$.

where $\|Cv\|_2 \leq \|v\|_2$ for all v.

Proof:
$C = \frac{1}{\lambda}(A - (1-\lambda)J)$.
Consider $v = u + w$, with $u = \alpha 1$ and $w \perp 1$.
$Cu = \frac{1}{\lambda}(A - (1-\lambda)J)u = \frac{1}{\lambda}(1-(1-\lambda))u = u$
$w' = Aw$, $|w'|_2^2 = w^T AAw = \lambda^2 |w|_2^2$.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2 \|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$$A = (1-\lambda)J + \lambda C.$$

where $\|Cv\|_2 \leq \|v\|_2$ for all v.

Proof:
$C = \frac{1}{\lambda}(A - (1-\lambda)J)$.
Consider $v = u + w$, with $u = \alpha 1$ and $w \perp 1$.
$Cu = \frac{1}{\lambda}(A - (1-\lambda)J)u = \frac{1}{\lambda}(1 - (1-\lambda))u = u$
$w' = Aw$, $|w'|_2^2 = w^T AAw = \lambda^2 |w|_2^2$.
$\implies \|Cw\|_2 = \frac{1}{\lambda}\|Aw\|_2 \leq \|w\|_2$.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2 \|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$$A = (1-\lambda)J + \lambda C.$$

where $\|Cv\|_2 \leq \|v\|_2$ for all v.

Proof:
$C = \frac{1}{\lambda}(A - (1-\lambda)J)$.
Consider $v = u + w$, with $u = \alpha 1$ and $w \perp 1$.
$Cu = \frac{1}{\lambda}(A - (1-\lambda)J)u = \frac{1}{\lambda}(1 - (1-\lambda))u = u$
$w' = Aw$, $|w'|_2^2 = w^T AAw = \lambda^2 |w|_2^2$.
$\implies \|Cw\|_2 = \frac{1}{\lambda}\|Aw\|_2 \leq \|w\|_2$. □

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2\|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$A = (1-\lambda)J + \lambda C$.

where $\|Cv\|_2 \leq \|v\|_2$ for all v.

Proof:
$C = \frac{1}{\lambda}(A-(1-\lambda)J)$.
Consider $v = u+w$, with $u = \alpha 1$ and $w \perp 1$.
$Cu = \frac{1}{\lambda}(A-(1-\lambda)J)u = \frac{1}{\lambda}(1-(1-\lambda))u = u$
$w' = Aw$, $|w'|_2^2 = w^T AAw = \lambda^2 |w|_2^2$.
$\implies \|Cw\|_2 = \frac{1}{\lambda}\|Aw\|_2 \leq \|w\|_2$. □
Remember: $|B|/n = \beta$.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2\|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$A = (1-\lambda)J + \lambda C$.

where $\|Cv\|_2 \leq \|v\|_2$ for all v.

Proof:
$C = \frac{1}{\lambda}(A-(1-\lambda)J)$.
Consider $v = u + w$, with $u = \alpha 1$ and $w \perp 1$.
$Cu = \frac{1}{\lambda}(A-(1-\lambda)J)u = \frac{1}{\lambda}(1-(1-\lambda))u = u$
$w' = Aw$, $|w'|_2^2 = w^T AAw = \lambda^2 |w|_2^2$.
$\implies \|Cw\|_2 = \frac{1}{\lambda}\|Aw\|_2 \leq \|w\|_2$. □
Remember: $|B|/n = \beta$.

$\hat{B}A = \hat{B}((1-\lambda)J + \lambda C)$

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \le \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2\|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$$A = (1-\lambda)J + \lambda C.$$

where $\|Cv\|_2 \le \|v\|_2$ for all v.

Proof:
$C = \frac{1}{\lambda}(A-(1-\lambda)J)$.
Consider $v = u+w$, with $u = \alpha 1$ and $w \perp 1$.
$Cu = \frac{1}{\lambda}(A-(1-\lambda)J)u = \frac{1}{\lambda}(1-(1-\lambda))u = u$
$w' = Aw$, $|w'|_2^2 = w^T AAw = \lambda^2 |w|_2^2$.
$\implies \|Cw\|_2 = \frac{1}{\lambda}\|Aw\|_2 \le \|w\|_2$. □
Remember: $|B|/n = \beta$.

$\hat{B}A = \hat{B}((1-\lambda)J + \lambda C)$ and $\|\hat{B}J\|_2 \le \sqrt{\beta}$

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2 \|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$A = (1-\lambda)J + \lambda C$.

where $\|Cv\|_2 \leq \|v\|_2$ for all v.

Proof:
$C = \frac{1}{\lambda}(A - (1-\lambda)J)$.
Consider $v = u + w$, with $u = \alpha 1$ and $w \perp 1$.
$Cu = \frac{1}{\lambda}(A-(1-\lambda)J)u = \frac{1}{\lambda}(1-(1-\lambda))u = u$
$w' = Aw$, $|w'|_2^2 = w^T AAw = \lambda^2 |w|_2^2$.
$\implies \|Cw\|_2 = \frac{1}{\lambda}\|Aw\|_2 \leq \|w\|_2$. □

Remember: $|B|/n = \beta$.

$\hat{B}A = \hat{B}((1-\lambda)J + \lambda C)$ and $\|\hat{B}J\|_2 \leq \sqrt{\beta}$ and $\|\hat{B}C\| \leq 1$.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2\|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$A = (1-\lambda)J + \lambda C$.

where $\|Cv\|_2 \leq \|v\|_2$ for all v.

Proof:
$C = \frac{1}{\lambda}(A - (1-\lambda)J)$.
Consider $v = u + w$, with $u = \alpha 1$ and $w \perp 1$.
$Cu = \frac{1}{\lambda}(A-(1-\lambda)J)u = \frac{1}{\lambda}(1-(1-\lambda))u = u$
$w' = Aw$, $|w'|_2^2 = w^T AAw = \lambda^2 |w|_2^2$.
$\implies \|Cw\|_2 = \frac{1}{\lambda}\|Aw\|_2 \leq \|w\|_2$. □

Remember: $|B|/n = \beta$.

$\hat{B}A = \hat{B}((1-\lambda)J + \lambda C)$ and $\|\hat{B}J\|_2 \leq \sqrt{\beta}$ and $\|\hat{B}C\| \leq 1$.
$\implies \|BA\|_2 \leq (1-\lambda)\sqrt{\beta} + \lambda$

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2 \|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$$A = (1-\lambda)J + \lambda C.$$

where $\|Cv\|_2 \leq \|v\|_2$ for all v.

Proof:
$C = \frac{1}{\lambda}(A - (1-\lambda)J)$.
Consider $v = u + w$, with $u = \alpha 1$ and $w \perp 1$.
$Cu = \frac{1}{\lambda}(A - (1-\lambda)J)u = \frac{1}{\lambda}(1-(1-\lambda))u = u$
$w' = Aw$, $|w'|_2^2 = w^T AAw = \lambda^2 |w|_2^2$.
$\implies \|Cw\|_2 = \frac{1}{\lambda}\|Aw\|_2 \leq \|w\|_2$. □

Remember: $|B|/n = \beta$.

$\hat{B}A = \hat{B}((1-\lambda)J + \lambda C)$ and $\|\hat{B}J\|_2 \leq \sqrt{\beta}$ and $\|\hat{B}C\| \leq 1$.
$\implies \|BA\|_2 \leq (1-\lambda)\sqrt{\beta} + \lambda$

Also, $\|\hat{B}\mathbf{1}\|_\mathbf{2} \leq \frac{\sqrt{\beta}}{\sqrt{\mathbf{n}}}$.

Bounding the 2-norm of A.

Def: $\|B\|_2$, is $max \frac{\|Bx\|_2}{\|x\|_2}$. $\|A+B\|_2 \le \|A\|_2 + \|B\|_2$. $\|AB\|_2 = \|A\|_2\|B\|_2$.

J scaled adjacency matrix of clique: $J_{i,j} = 1/n$.

Claim: If A is scaled adjacency matrix for λ expander.

$$A = (1-\lambda)J + \lambda C.$$

where $\|Cv\|_2 \le \|v\|_2$ for all v.

Proof:
$C = \frac{1}{\lambda}(A - (1-\lambda)J)$.
Consider $v = u + w$, with $u = \alpha 1$ and $w \perp 1$.
$Cu = \frac{1}{\lambda}(A-(1-\lambda)J)u = \frac{1}{\lambda}(1-(1-\lambda))u = u$
$w' = Aw$, $|w'|_2^2 = w^T AAw = \lambda^2 |w|_2^2$.
$\implies \|Cw\|_2 = \frac{1}{\lambda}\|Aw\|_2 \le \|w\|_2$. □

Remember: $|B|/n = \beta$.

$\hat{B}A = \hat{B}((1-\lambda)J + \lambda C)$ and $\|\hat{B}J\|_2 \le \sqrt{\beta}$ and $\|\hat{B}C\| \le 1$.
$\implies \|BA\|_2 \le (1-\lambda)\sqrt{\beta} + \lambda$

Also, $\|\hat{B}\mathbf{1}\|_2 \le \frac{\sqrt{\beta}}{\sqrt{n}}$. $\implies \|(\hat{B}A)^\ell \hat{B}\mathbf{1}\|_2 \le \frac{((1-\lambda)\sqrt{\beta}+\lambda)^\ell \sqrt{\beta}}{\sqrt{N}}$

Also.

Flip k coins: don't get heads with probability $(1/2)^k$.

Also.

Flip k coins: don't get heads with probability $(1/2)^k$.

Analagous statement for expanders:

Also.

Flip k coins: don't get heads with probability $(1/2)^k$.

Analagous statement for expanders: $(f(\lambda, 1/2))^k$, where $f(\lambda, 1/2)) < 1$.

Also.

Flip k coins: don't get heads with probability $(1/2)^k$.

Analagous statement for expanders: $(f(\lambda,1/2))^k$, where $f(\lambda,1/2)) < 1$.

Flip k coins: get roughly $k/2$ heads.

Also.

Flip k coins: don't get heads with probability $(1/2)^k$.

Analagous statement for expanders: $(f(\lambda,1/2))^k$, where $f(\lambda,1/2)) < 1$.

Flip k coins: get roughly $k/2$ heads.

Something analagous for walk in expanders.

Summary.

Eigenvectors for hypercubes.

Summary.

Eigenvectors for hypercubes.

Tight example for LHI of Cheeger. Eigenvectors for cycle.

Summary.

Eigenvectors for hypercubes.
Tight example for LHI of Cheeger. Eigenvectors for cycle.
Tight example for RHI of Cheeger.

Summary.

Eigenvectors for hypercubes.
  Tight example for LHI of Cheeger. Eigenvectors for cycle.
  Tight example for RHI of Cheeger.

Random Walks and Sampling.

Summary.

Eigenvectors for hypercubes.
  Tight example for LHI of Cheeger. Eigenvectors for cycle.
  Tight example for RHI of Cheeger.

Random Walks and Sampling.

Eigenvectors, Isoperimetry of Volume, Mixing.

Summary.

Eigenvectors for hypercubes.
  Tight example for LHI of Cheeger. Eigenvectors for cycle.
  Tight example for RHI of Cheeger.

Random Walks and Sampling.

Eigenvectors, Isoperimetry of Volume, Mixing.

Partial Order Application.