Random Walks as a Stable Analogue of Eigenvectors

(with Applications to Nearly-Linear-Time Graph Partitioning)

Lorenzo Orecchia, MIT Math

Based on joint works with Michael Mahoney (Stanford), Sushant Sachdeva (Yale) and Nisheeth Vishnoi (MSR India).

Why Spectral Algorithms for Graph Problems …

… in practice?

- Simple to implement
- Can exploit very efficient linear algebra routines
- Perform well in practice for many problems

… in theory?

- Connections between spectral and combinatorial objects
- Connections to Markov Chains and Probability Theory
- Intuitive geometric viewpoint

RECENT ADVANCES:

Fast algorithms for fundamental combinatorial problems

<u>rely on spectral and optimization ideas</u>

Spectral Algorithms for Graph Partitioning

Spectral algorithms are widely used in many graph-partitioning applications: clustering, image segmentation, community-detection, etc.

CLASSICAL VIEW:

- Based on Cheeger's Inequality

- Eigenvectors sweep-cuts reveal sparse cuts in the graph

Spectral Algorithms for Graph Partitioning

Spectral algorithms are widely used in many graph-partitioning applications:
clustering, image segmentation, community-detection, etc.

CLASSICAL VIEW:

- Based on Cheeger's Inequality
- Eigenvectors sweep-cuts reveal sparse cuts in the graph

NEW TREND:

- Random walk vectors replace eigenvectors:
  - Fast Algorithms for Graph Partitioning
  - Local Graph Partitioning
  - Real Network Analysis
- Different random walks: PageRank, Heat-Kernel, etc.

Why Random Walks? A Practitioner's View

Advantages of Random Walks:

1) Quick approximation to eigenvector in massive graphs

A = adjacency matrix

D = diagonal degree matrix

$W = AD^{-1}$ = natural random walk matrix

$L = D - A$ = Laplacian matrix

Second Eigenvector of the Laplacian can be computed by iterating W :

For random y_0 s.t. $y_0^T D^{-1} 1 = 0$, compute

$$D^{-1} W^t y_0$$

Why Random Walks? A Practitioner's View

Advantages of Random Walks:

1) Quick approximation to eigenvector in massive graphs

A = adjacency matrix

D = diagonal degree matrix

$W = AD^{-1}$ = natural random walk matrix

$L = D - A$ = Laplacian matrix

Second Eigenvector of the Laplacian can be computed by iterating W :

For random y_0 s.t. $y_0^T D^{-1} 1 = 0$, compute

$$D^{-1} W^t y_0$$

In the limit, $x_2(L) = \lim_{t \to \infty} \frac{D^{-1} W^t y_0}{||W^t y_0||_{D^{-1}}}$.

Why Random Walks? A Practitioner's View

Advantages of Random Walks:

1) Quick approximation to eigenvector in massive graphs

A = adjacency matrix

D = diagonal degree matrix

$W = AD^{-1}$ = natural random walk matrix

$L = D - A$ = Laplacian matrix

Second Eigenvector of the Laplacian can be computed by iterating W :

For random y_0 s.t. $y_0^T D^{-1} 1 = 0$, compute

$$D^{-1} W^t y_0$$

In the limit, $x_2(L) = \lim_{t \to \infty} \frac{D^{-1} W^t y_0}{||W^t y_0||_{D^{-1}}}$.

Heuristic: For massive graphs, pick t as large as computationally affordable.

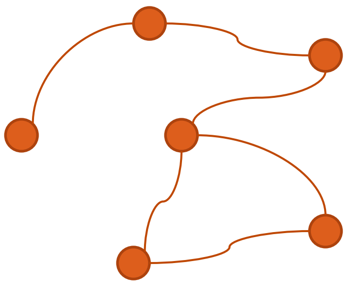
Why Random Walks? A Practitioner's View

Advantages of Random Walks:

1) Quick approximation to eigenvector in massive graphs

2) <u>Statistical robustness</u>

   Real-world graphs are noisy



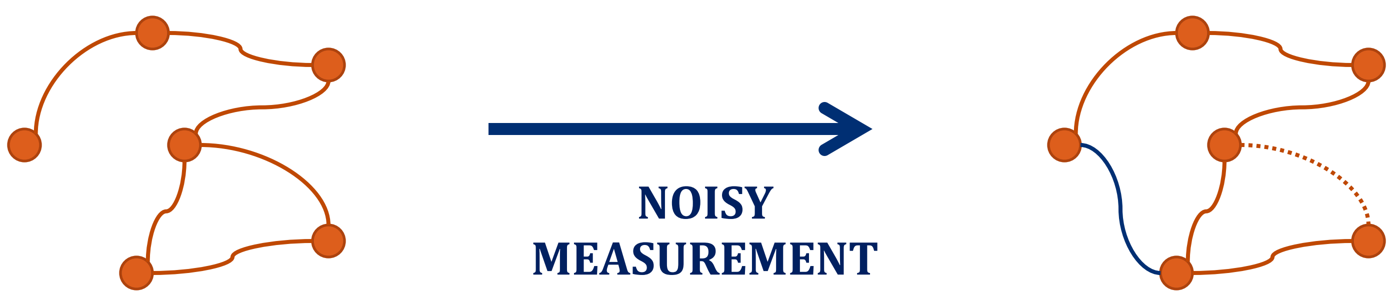
GROUND TRUTH GRAPH

Why Random Walks? A Practitioner's View

Advantages of Random Walks:

1) Quick approximation to eigenvector in massive graphs
2) <u>Statistical robustness</u>

Real-world graphs are noisy



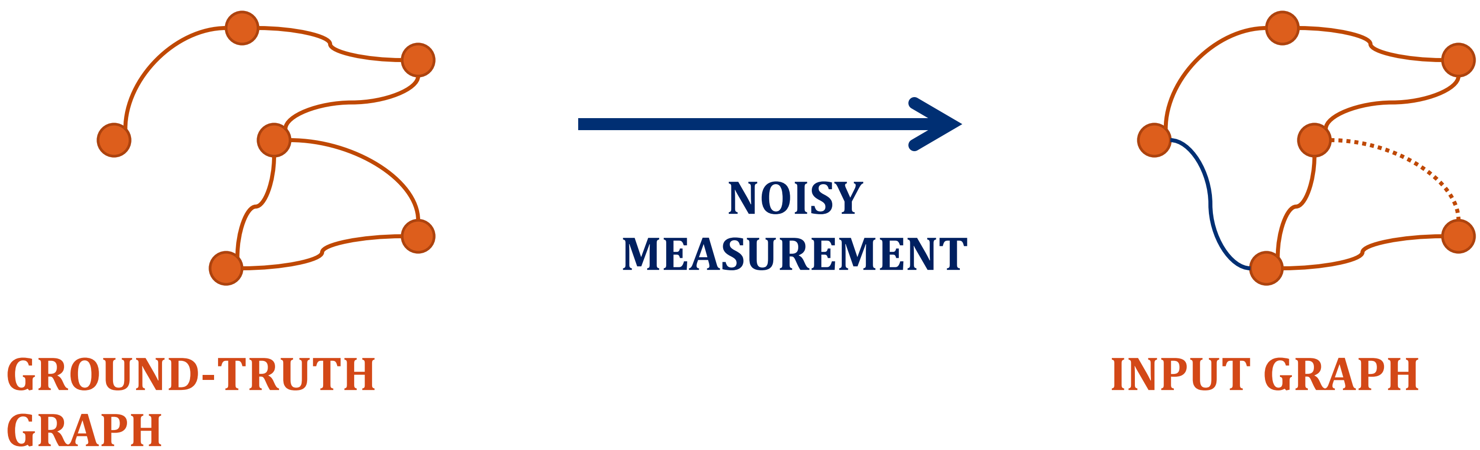
<u>GOAL</u>: estimate eigenvector of ground-truth graph.

Why Random Walks? A Practitioner's View

Advantages of Random Walks:

1) Quick approximation to eigenvector in massive graphs

2) Statistical robustness



GOAL: estimate eigenvector of ground-truth graph.

OBSERVATION: eigenvector of input graph can have very large variance, as it can be very sensitive to noise

RANDOM-WALK VECTORS provide better, more stable estimates.

This Talk

QUESTION:

Why random-walk vectors in the design of fast algorithms?

This Talk

QUESTION:

Why random-walk vectors in the design of fast algorithms?

ANSWER: Stable, **regularized** version of the eigenvector

This Talk

QUESTION:

Why random-walk vectors in the design of fast algorithms?

ANSWER: Stable, **regularized** version of the eigenvector

GOALS OF THIS TALK:

- Show optimization perspective on why random walks arise

- Application to nearly-linear-time balanced graph partitioning

Random Walks
as Regularized Eigenvectors

What is Regularization?

Regularization is a fundamental technique in optimization

OPTIMIZATION PROBLEM $\longrightarrow$ <u>WELL-BEHAVED</u> OPTIMIZATION PROBLEM

- Stable optimum
- Unique optimal solution
- Smoothness conditions

…

What is Regularization?

Regularization is a fundamental technique in optimization

OPTIMIZATION PROBLEM → <u>WELL-BEHAVED</u> OPTIMIZATION PROBLEM

$$\min_{x \in H} L(x) \qquad\qquad \min_{x \in H} L(x) + \lambda \cdot F(x)$$

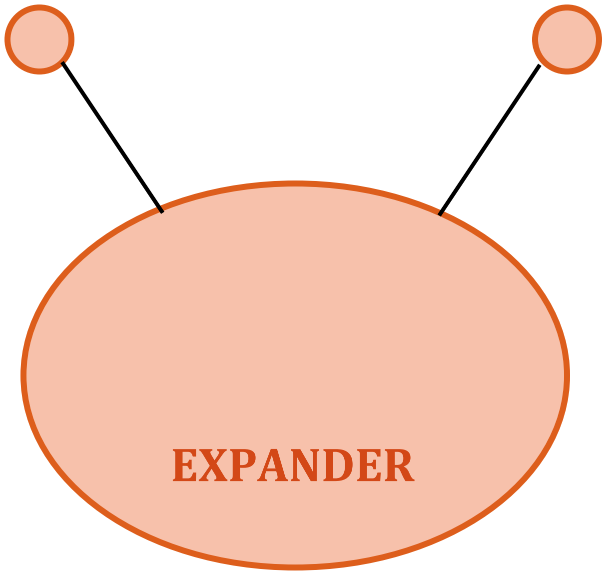
Parameter $\lambda > 0$

Regularizer F

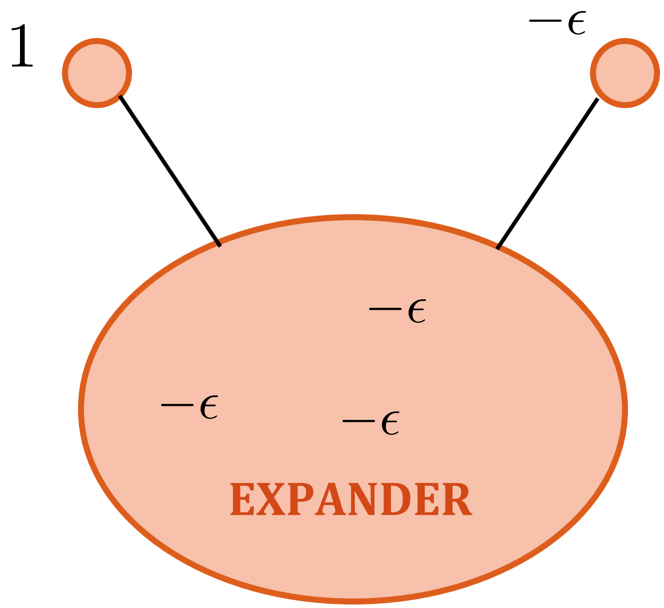
Benefits of Regularization in Learning and Statistics:

- Increases stability
- Decreases sensitivity to random noise
- Prevents overfitting

Instability of Eigenvector

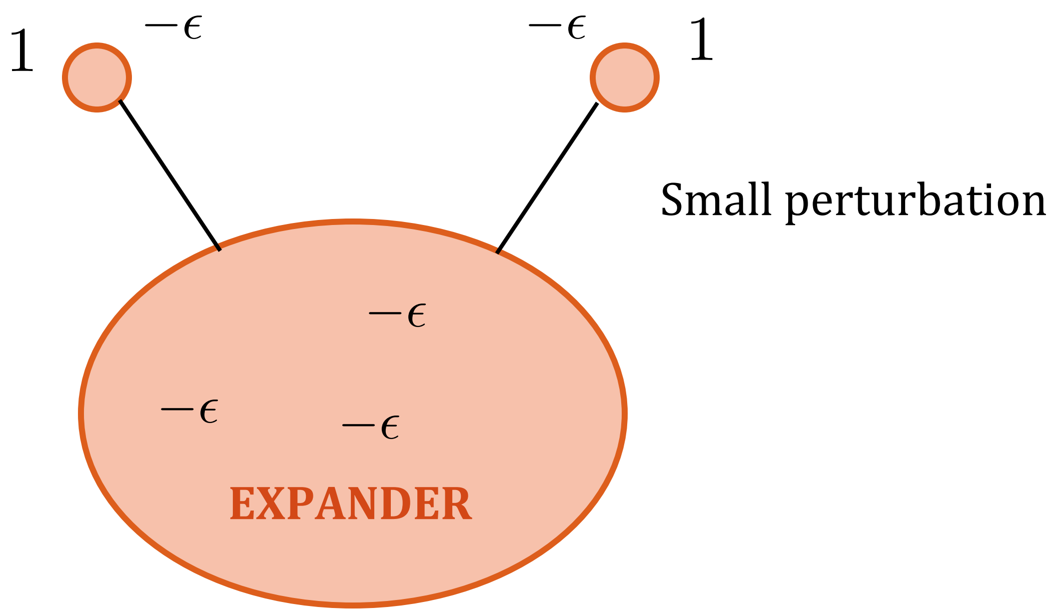


Instability of Eigenvector



Current eigenvector

Instability of Eigenvector



Current eigenvector

Eigenvector Changes Completely!

The Laplacian Eigenvalue Problem

Quadratic Formulation

$$\frac{1}{d} \min x^T L x$$

$$\text{s.t. } ||x||_2 = 1$$

$$x^T 1 = 0$$

For simplicity, take G to be d-regular.

The Laplacian Eigenvalue Problem

Quadratic Formulation

SDP Formulation

$$\frac{1}{d}\min x^T L x \quad\longleftrightarrow\quad \frac{1}{d}\min \quad L \bullet X$$

$$\text{s.t. } ||x||_2 = 1 \qquad\qquad \text{s.t.} \quad I \bullet X = 1$$

$$x^T 1 \quad = 0 \qquad\qquad 11^T \bullet X = 0$$

$$X \succeq 0$$

The Laplacian Eigenvalue Problem

Quadratic Formulation

SDP Formulation

$$\frac{1}{d}\min x^T L x \qquad\longleftrightarrow\qquad \frac{1}{d}\min \quad L \bullet X$$

$$\text{s.t. } ||x||_2 = 1 \qquad\qquad \text{s.t.} \quad I \bullet X = 1$$

$$x^T 1 \;= 0 \qquad\qquad 11^T \bullet X = 0$$

$$X \succeq 0$$

Programs have same optimum. Take optimal solution

$$X^* = x^*(x^*)^T$$

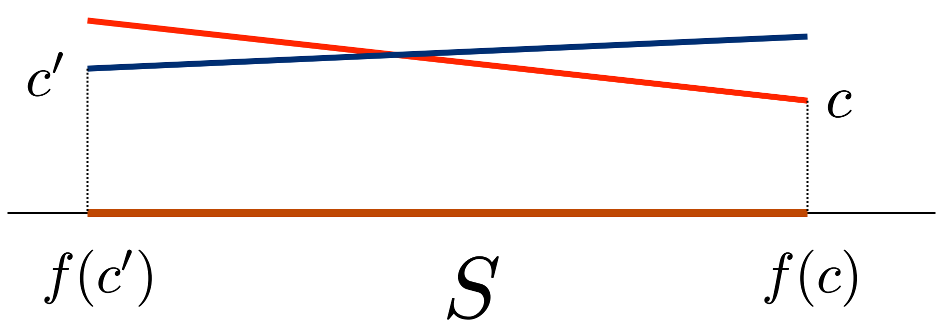
Instability of Linear Optimization

Consider a convex set $S \subset R^n$ and a linear optimization problem:

$$f(c) = \arg\min_{x \in S} c^T x$$

The optimal solution $f(c)$ may be very unstable under perturbation of c :

$$\|c' - c\| \le \delta \quad \text{and} \quad \|f(c') - f(c)\| >> \delta$$



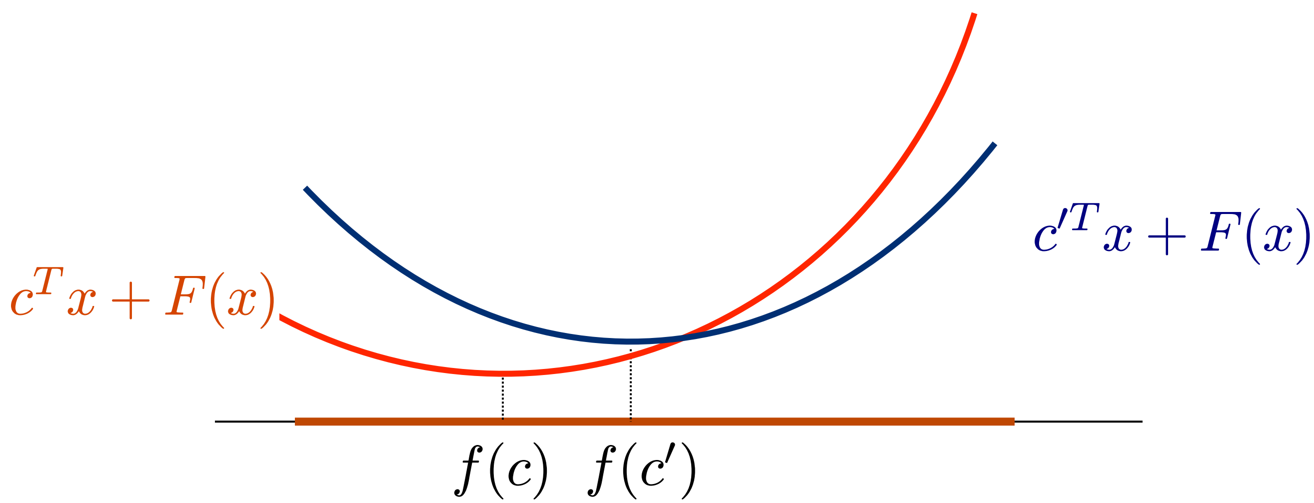
Regularization Helps Stability

Consider a convex s$S \subset R^n$ and a **regularized** linear optimization problem

$$f(c) = \arg\min_{x \in S} c^T x + F(x)$$

where F is σ-strongly convex.

Then: $\|c' - c\| \leq \delta$ implies $\|f(c) - f(c')\| \leq \dfrac{\delta}{\sigma}$



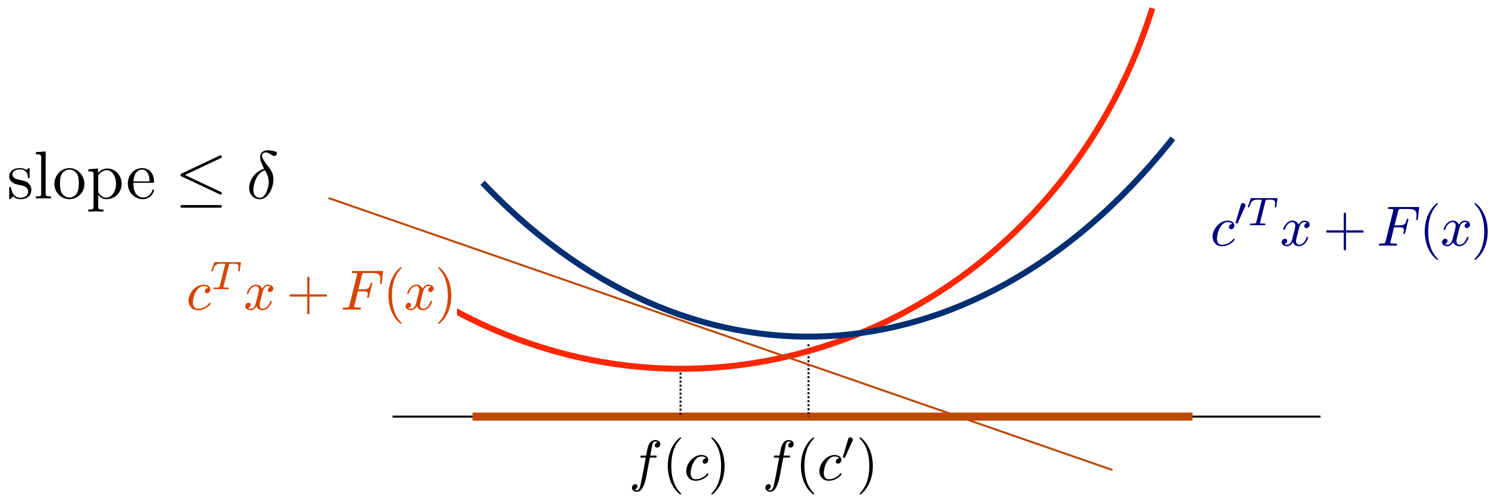
Regularization Helps Stability

Consider a convex s $S \subset R^n$ and a **regularized** linear optimization problem

$$f(c) = \arg\min_{x \in S} c^T x + F(x)$$

where F is σ-strongly convex.

Then: $\|c' - c\| \leq \delta$ implies $\|f(c) - f(c')\| \leq \frac{\delta}{\sigma}$



Regularized Spectral Optimization

SDP Formulation

$$
\begin{aligned}
\frac{1}{d}\min \quad & L \bullet X \\
\text{s.t.} \quad & \left.\begin{aligned} I \bullet X &= 1 \\ 11^T \bullet X &= 0 \\ X &\succeq 0 \end{aligned}\right\} \text{Density Matrix}
\end{aligned}
$$

Eigenvector decomposition of X:

$$
X = \sum p_i v_i v_i^T \quad \left\{ \begin{aligned} &\forall i, p_i \geq 0, \\ &\sum p_i = 1, \\ &\forall i, v_i^T 1 = 0. \end{aligned} \right.
$$

Eigenvalues of X define probability distribution

Regularized Spectral Optimization

SDP Formulation

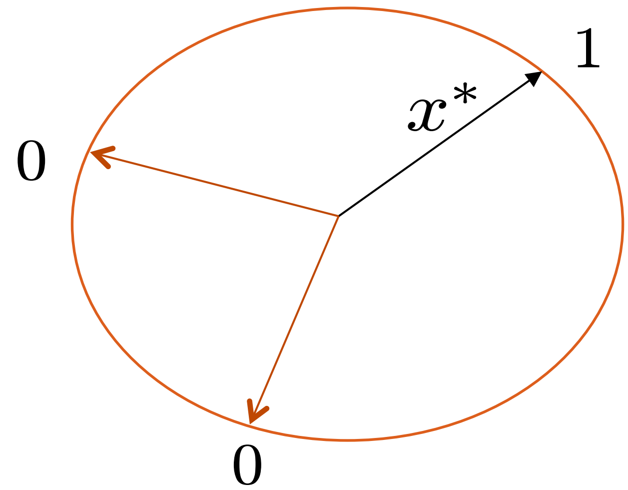
$$\frac{1}{d} \min L \bullet X$$

$$\text{s.t.} \quad \left.\begin{aligned} I \bullet X &= 1 \\ J \bullet X &= 0 \\ X &\succeq 0 \end{aligned}\right\} \text{Density Matrix}$$

Eigenvalues of X define probability distribution

$$X^* = x^* (x^*)^T$$

TRIVIAL DISTRIBUTION



Regularized Spectral Optimization

$$\frac{1}{d}\min \quad L \bullet X + \boxed{\eta \cdot F(X)}$$

Regularizer F
Parameter η

$$\text{s.t.} \quad I \bullet X = 1$$

$$11^T \bullet X = 0$$

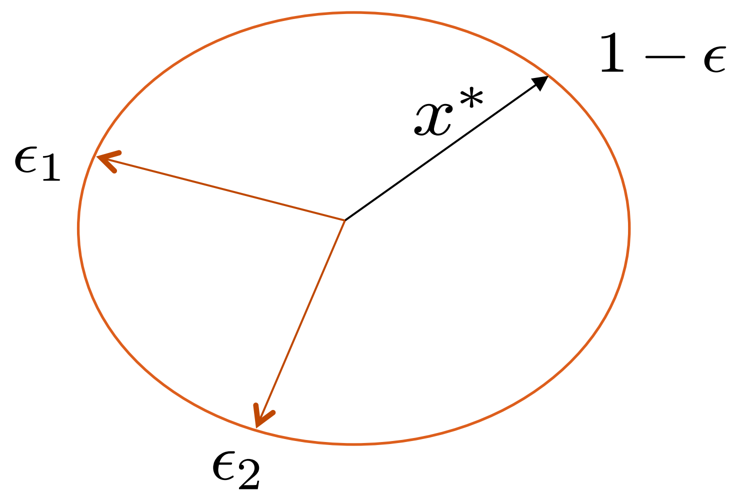
$$X \succeq 0$$

The regularizer F forces the distribution of eigenvalues of X to be non-trivial

$$X^* = x^*(x^*)^T$$

REGULARIZATION

$$X^* = \sum p_i v_i v_i^T$$



Regularizers

Regularizers are SDP-versions of common regularizers

- von Neumann Entropy

$$F_H(X) = \mathrm{Tr}(X \log X) = \sum p_i \log p_i$$

- p-Norm, $p > 1$

$$F_p(X) = \frac{1}{p} ||X||_p^p = \frac{1}{p} \mathrm{Tr}(X^p) = \frac{1}{p} \sum p_i^p$$

- And more, e.g. log-determinant.

Our Main Result

Regularized SDP

$$\frac{1}{d}\min \quad L \bullet X + \eta \cdot F(X)$$

$$\text{s.t.} \quad I \bullet X = 1$$

$$J \bullet X = 0$$

$$X \succeq 0$$

RESULT:

Explicit correspondence between **regularizers and random walks**

REGULARIZER

OPTIMAL SOLUTION OF REGULARIZED PROGRAM

$$F = F_H \xrightarrow[\text{Entropy}]{} X^\star \propto H_G^t$$

where t depends on η

$$F = F_p \xrightarrow[p\text{-Norm}]{} X^\star \propto (qI + (1-q)W)^{\frac{1}{p-1}}$$

where q depends on η

Our Main Result

Regularized SDP

$$\frac{1}{d}\min \quad L \bullet X + \eta \cdot F(X)$$

$$\text{s.t.} \quad I \bullet X = 1$$

$$J \bullet X = 0$$

$$X \succeq 0$$

RESULT:

Explicit correspondence between **regularizers and random walks**

REGULARIZER

OPTIMAL SOLUTION OF REGULARIZED PROGRAM

$F = F_H \xrightarrow{\text{Entropy}}$ $X^\star \propto H_G^t$ where t depends on η

HEAT-KERNEL

$F = F_p \xrightarrow{p\text{-Norm}}$ $X^\star \propto (qI + (1-q)W)^{\frac{1}{p-1}}$

LAZY RANDOM WALK where q depends on η

Background: Heat-Kernel Random Walk

For simplicity, take G to be **d-regular**.

• The Heat-Kernel Random Walk is a Continuous-Time Markov Chain over V, modeling the diffusion of heat along the edges of G.

• Transitions take place in continuous time t, with an exponential distribution.

$$\frac{\partial p(t)}{\partial t} = -L\frac{p(t)}{d}$$

$$p(t) = e^{-\frac{t}{d}L}p(0)$$

• The Heat Kernel can be interpreted as Poisson distribution over number of steps of the natural random walk $W=AD^{-1}$:

$$e^{-\frac{t}{d}L} = e^{-t}\sum_{k=1}^{\infty}\frac{t^k}{k!}W^k$$

Background: Heat-Kernel Random Walk

For simplicity, take G to be **d-regular**.

• The Heat-Kernel Random Walk is a Continuous-Time Markov Chain over V, modeling the diffusion of heat along the edges of G.

• Transitions take place in continuous time t, with an exponential distribution.

$$\frac{\partial p(t)}{\partial t} = -L\frac{p(t)}{d}$$

Notation

$$p(t) = e^{-\frac{t}{d}L}p(0) =: \boxed{H_G^t} \; p(0)$$

• The Heat Kernel can be interpreted as Poisson distribution over number of steps of the natural random walk $W{=}AD^{-1}$:

$$e^{-\frac{t}{d}L} = e^{-t}\sum_{k=1}^{\infty}\frac{t^k}{k!}W^k$$

Heat Kernel Walk: Stability Analysis

Consider a convex s$S \subset R^n$ and a **regularized** linear optimization problem

$$f(c) = \arg\min_{x \in S} c^T x + F(x)$$

where F is σ-strongly convex.

Then: $\|c' - c\| \leq \delta$ implies $\|f(c) - f(c')\| \leq \dfrac{\delta}{\sigma}$

Heat Kernel Walk: Stability Analysis

Consider a convex set $S \subset R^n$ and a **regularized** linear optimization problem

$$f(c) = \arg\min_{x \in S} c^T x + F(x)$$

where F is σ-strongly convex.

Then: $\|c' - c\| \leq \delta$ implies $\|f(c) - f(c')\| \leq \dfrac{\delta}{\sigma}$

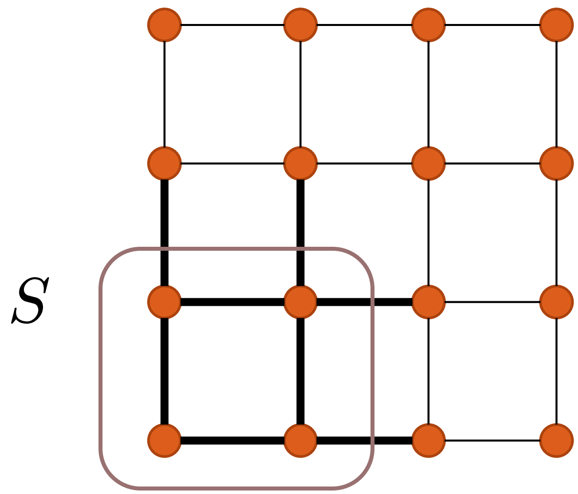
Analogous statement for Heat Kernel:

$$\|G' - G\|_\infty \leq \delta \quad \text{implies} \quad \left\| \frac{H^\tau_{G'}}{I \bullet H^\tau_{G'}} - \frac{H^\tau_G}{I \bullet H^\tau_G} \right\|_1 \leq \tau \cdot \delta$$

Applications to Graph Partitioning: Nearly-Linear-Time Balanced Cut

Partitioning Graphs - Conductance

Undirected unweighted $G = (V, E), |V| = n, |E| = m$



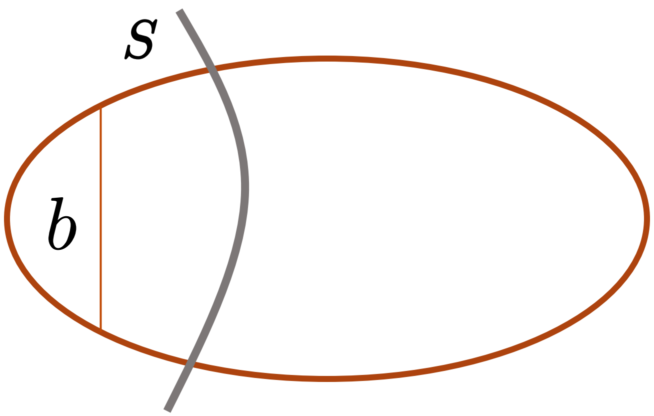
Conductance of $S \subseteq V$

$$\phi(S) = \frac{|E(S,\bar{S})|}{\min\{\mathrm{vol}(S),\mathrm{vol}(\bar{S})\}}$$

Partitioning Graphs – Balanced Cut

NP-HARD DECISION PROBLEM

Does G have a <u>b-balanced</u> cut of conductance < γ ?



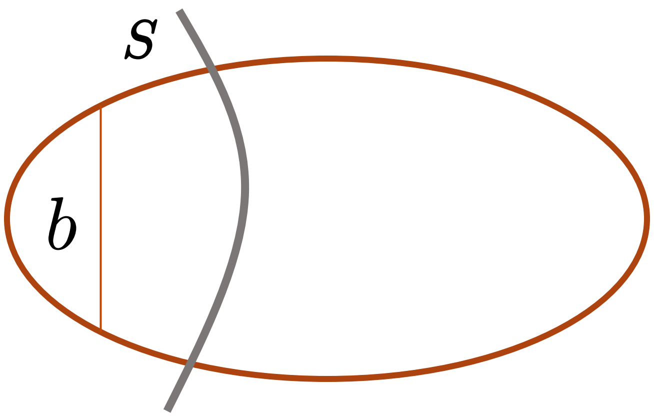
$$\phi(S) < \gamma$$

$$\tfrac{1}{2} > \frac{\mathrm{vol}(S)}{\mathrm{vol}(V)} > b$$

Partitioning Graphs – Balanced Cut

NP-HARD DECISION PROBLEM

Does G have a <u>b-balanced</u> cut of conductance < γ ?



$$\phi(S) < \gamma$$

$$\tfrac{1}{2} > \frac{\mathrm{vol}(S)}{\mathrm{vol}(V)} > b$$

- Important primitive for many **recursive** algorithms.
- Applications to **clustering** and **graph decomposition**.

Spectral Approximation Algorithms

Does G have a *b*-balanced cut of conductance < γ ?

Algorithm	Method	Distinguishes $\geq$ γ and	Running Time
Recursive Eigenvector	Spectral	$O(\sqrt{\gamma})$	$\tilde{O}(mn)$
[Spielman, Teng '04]	Local Random Walks	$O\left(\sqrt{\gamma \log^3 n}\right)$	$\tilde{O}\left(\frac{m}{\gamma^2}\right)$
[Andersen, Chung, Lang '07]	Local Random Walks	$O\left(\sqrt{\gamma \log n}\right)$	$\tilde{O}\left(\frac{m}{\gamma}\right)$
[Andersen, Peres '09]	Local Random Walks	$O\left(\sqrt{\gamma \log n}\right)$	$\tilde{O}\left(\frac{m}{\sqrt{\gamma}}\right)$

Spectral Approximation Algorithms

Does G have a *b*-balanced cut of conductance < γ ?

Algorithm	Method	Distinguishes $\geq \gamma$ and	Running Time
Recursive Eigenvector	Spectral	$O(\sqrt{\gamma})$	$\tilde{O}(mn)$
[Spielman, Teng '04]	Local Random Walks	$O\left(\sqrt{\gamma \log^3 n}\right)$	$\tilde{O}\left(\frac{m}{\gamma^2}\right)$
[Andersen, Chung, Lang '07]	Local Random Walks	$O\left(\sqrt{\gamma \log n}\right)$	$\tilde{O}\left(\frac{m}{\gamma}\right)$
[Andersen, Peres '09]	Local Random Walks	$O\left(\sqrt{\gamma \log n}\right)$	$\tilde{O}\left(\frac{m}{\sqrt{\gamma}}\right)$
[Orecchia, Sachdeva, Vishnoi '12]	Random Walks	$O(\sqrt{\gamma})$	$\tilde{O}\left(m\right)$

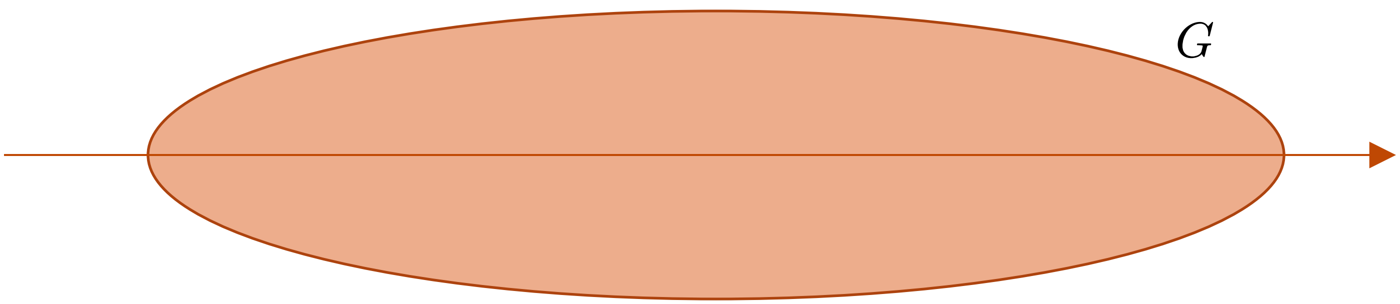
Recursive Eigenvector Algorithm

INPUT: (G, b, γ) **DECISION:** does there exists b-balanced S with $\phi(S) < \gamma$?

Recursive Eigenvector Algorithm

INPUT: (G, b, γ) **DECISION:** does there exists b-balanced S with $\phi(S) < \gamma$?

- Compute eigenvector of G and corresponding Laplacian eigenvalue λ_2

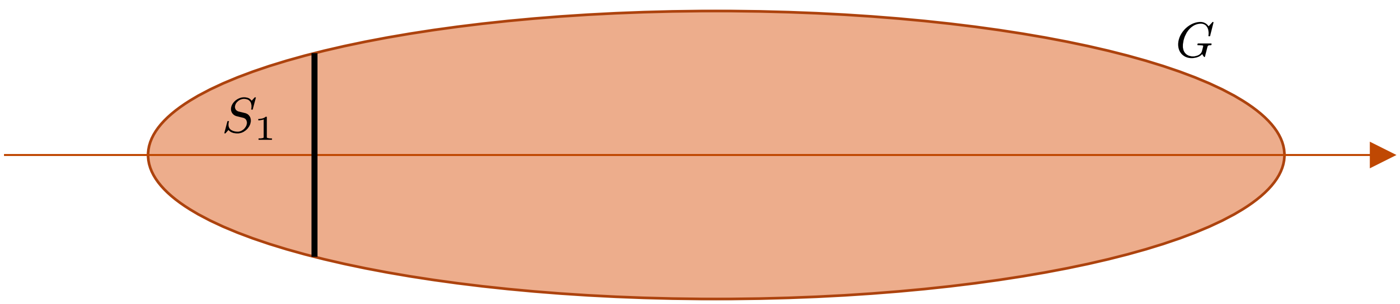


Recursive Eigenvector Algorithm

INPUT: (G, b, γ) **DECISION:** does there exists b-balanced S with $\phi(S) < \gamma$?

- Compute eigenvector of G and corresponding Laplacian eigenvalue λ_2
- If $\lambda_2 \geq \gamma$, output **NO**. Otherwise, sweep eigenvector to find S_1 such that

$$\phi(S_1) \leq O(\sqrt{\gamma})$$



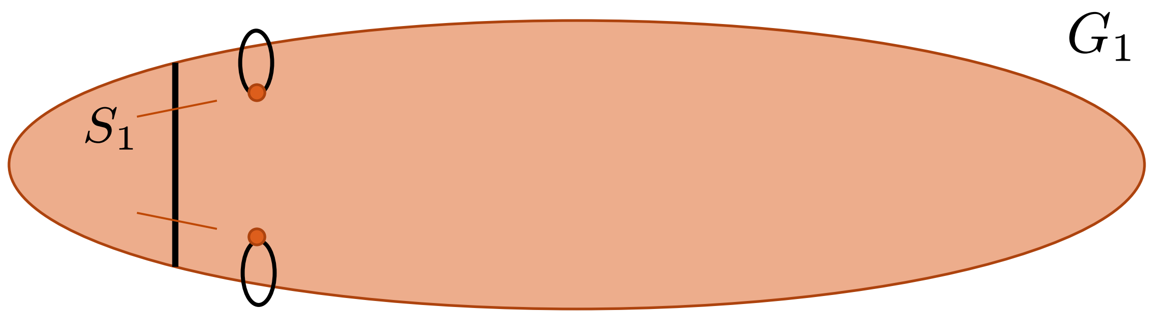
Recursive Eigenvector Algorithm

INPUT: (G, b, γ) **DECISION:** does there exists b-balanced S with $\phi(S) < \gamma$?

- Compute eigenvector of G and corresponding Laplacian eigenvalue λ_2
- If $\lambda_2 \geq \gamma$, output **NO**. Otherwise, sweep eigenvector to find S_1 such that

$$\phi(S_1) \leq O(\sqrt{\gamma})$$

- If S_1 is $(b/2)$-balanced. Output S_1. Otherwise, consider the graph G_1 induced by G on V-S_1 with self-loops replacing the edges going to S_1.



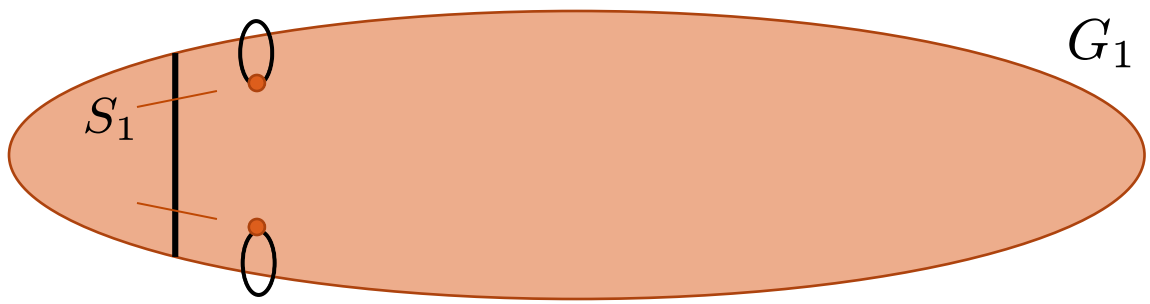
Recursive Eigenvector Algorithm

INPUT: (G, b, γ) **DECISION:** does there exists b-balanced S with $\phi(S) < \gamma$?

- Compute eigenvector of G and corresponding Laplacian eigenvalue λ_2
- If $\lambda_2 \geq \gamma$, output **NO**. Otherwise, sweep eigenvector to find S_1 such that

$$\phi(S_1) \leq O(\sqrt{\gamma})$$

- If S_1 is $(b/2)$-balanced. Output S_1. Otherwise, consider the graph G_1 induced by G on V-S_1 with self-loops replacing the edges going to S_1.
- Recurse on G_1.



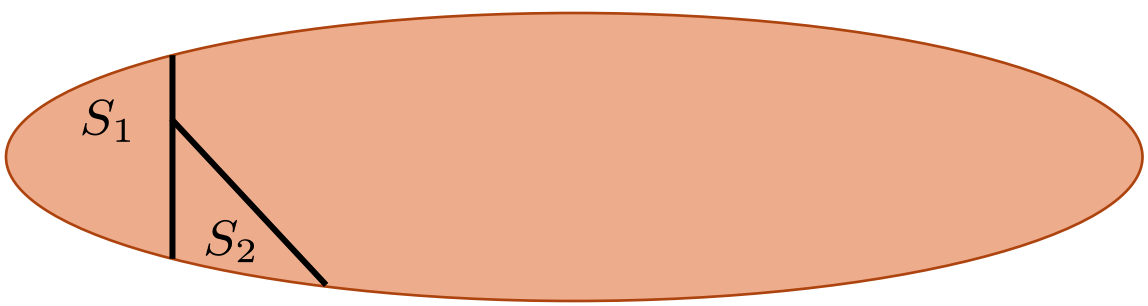
Recursive Eigenvector Algorithm

INPUT: (G, b, γ) **DECISION:** does there exists b-balanced S with $\phi(S) < \gamma$?

- Compute eigenvector of G and corresponding Laplacian eigenvalue λ_2
- If $\lambda_2 \geq \gamma$, output **NO**. Otherwise, sweep eigenvector to find S_1 such that

$$\phi(S_1) \leq O(\sqrt{\gamma})$$

- If S_1 is $(b/2)$-balanced. Output S_1. Otherwise, consider the graph G_1 induced by G on V-S_1 with self-loops replacing the edges going to S_1.
- Recurse on G_1.



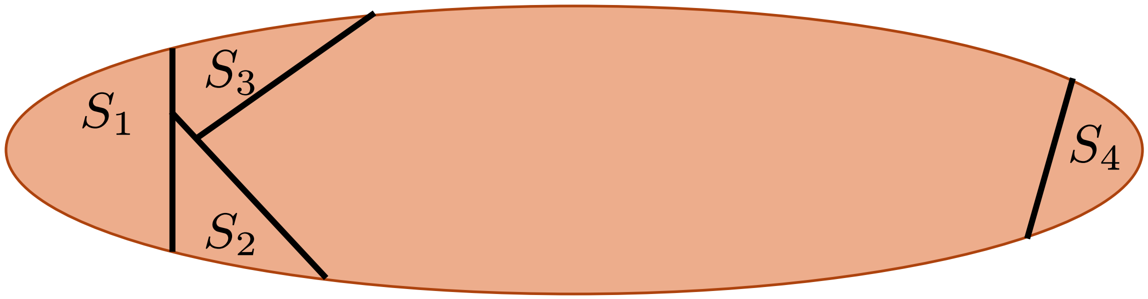
Recursive Eigenvector Algorithm

INPUT: (G, b, γ) **DECISION:** does there exists b-balanced S with $\phi(S) < \gamma$?

- Compute eigenvector of G and corresponding Laplacian eigenvalue λ_2
- If $\lambda_2 \geq \gamma$, output **NO**. Otherwise, sweep eigenvector to find S_1 such that

$$\phi(S_1) \leq O(\sqrt{\gamma})$$

- If S_1 is $(b/2)$-balanced. Output S_1. Otherwise, consider the graph G_1 induced by G on V-S_1 with self-loops replacing the edges going to S_1.
- Recurse on G_1.



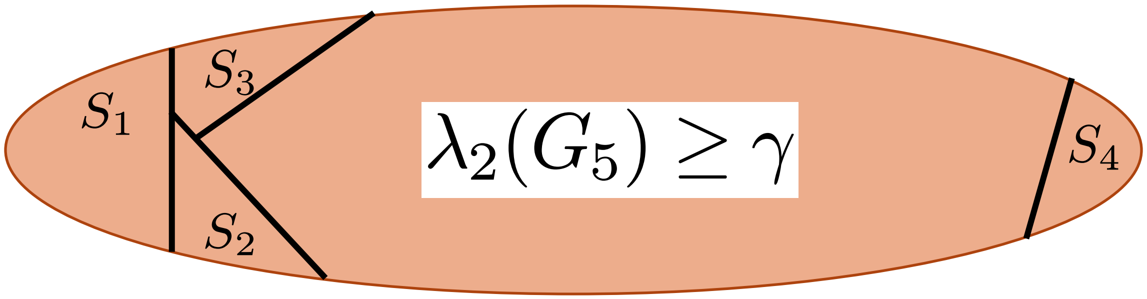
Recursive Eigenvector Algorithm

INPUT: (G, b, γ) **DECISION:** does there exists b-balanced S with $\phi(S) < \gamma$?

- Compute eigenvector of G and corresponding Laplacian eigenvalue λ_2
- If $\lambda_2 \geq \gamma$, output **NO**. Otherwise, sweep eigenvector to find S_1 such that

$$\phi(S_1) \leq O(\sqrt{\gamma})$$

- If S_1 is $(b/2)$-balanced. Output S_1. Otherwise, consider the graph G_1 induced by G on V-S_1 with self-loops replacing the edges going to S_1.
- Recurse on G_1.



LARGE INDUCED EXPANDER = **NO-CERTIFICATE**

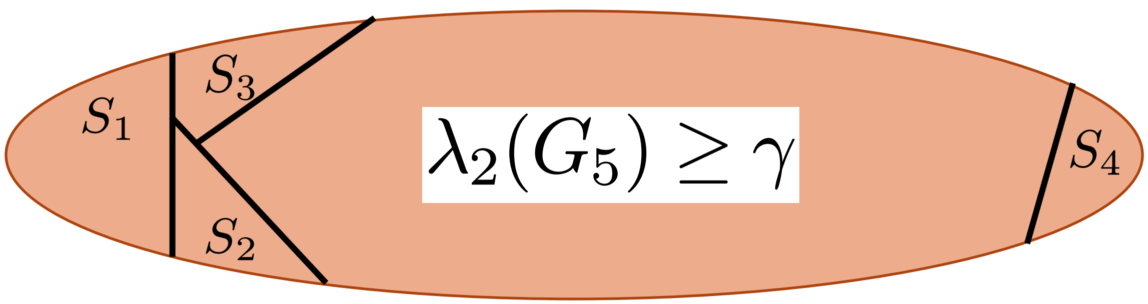
Recursive Eigenvector Algorithm

INPUT: (G, b, γ) **DECISION:** does there exists b-balanced S with $\phi(S) < \gamma$?

- Compute eigenvector of G and corresponding Laplacian eigenvalue λ_2
- If $\lambda_2 \geq \gamma$, output **NO**. Otherwise, sweep eigenvector to find S_1 such that

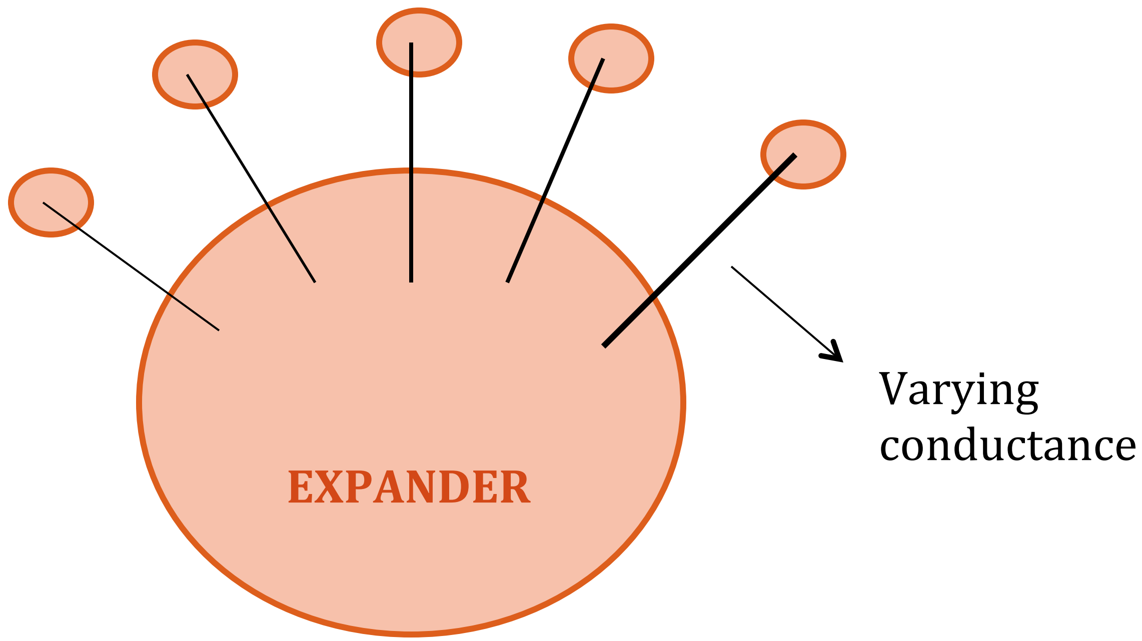
$$\phi(S_1) \leq O(\sqrt{\gamma})$$

- If S_1 is $(b/2)$-balanced. Output S_1. Otherwise, consider the graph G_1 induced by G on V-S_1 with self-loops replacing the edges going to S_1.
- Recurse on G_1.



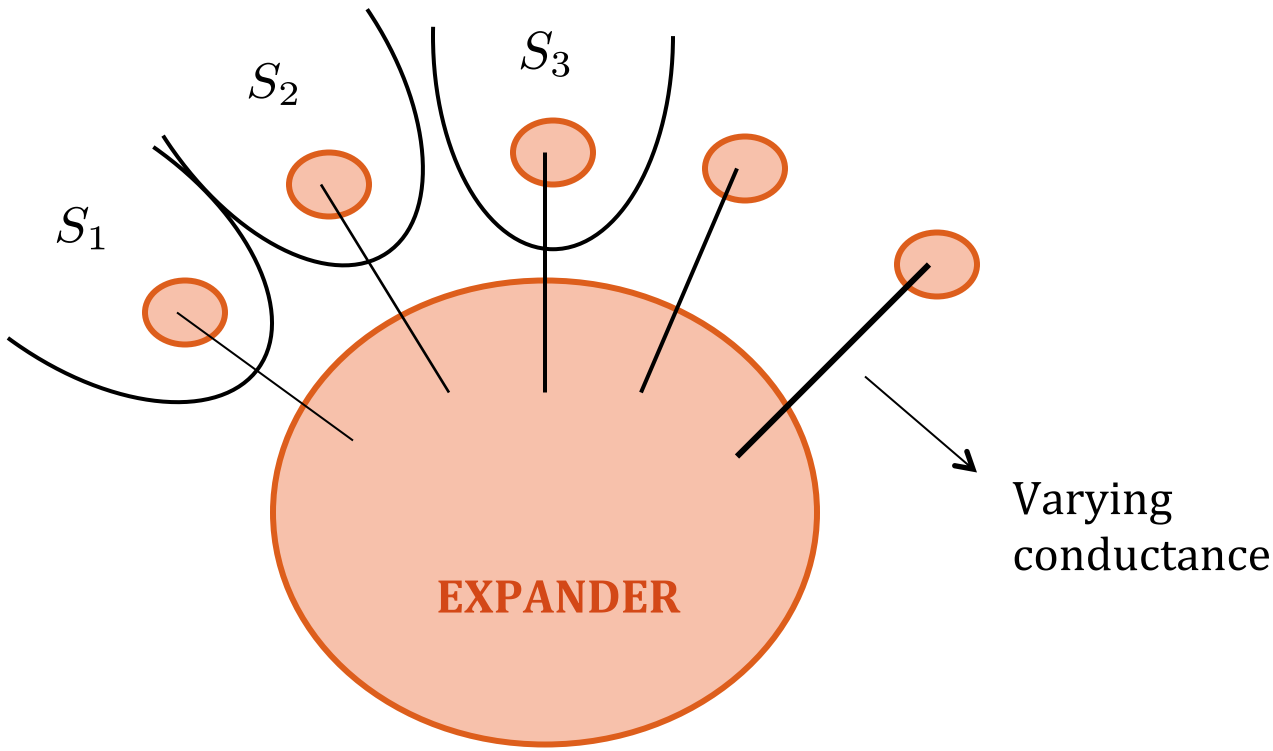
RUNNING TIME: $\tilde{O}(m)$ per iteration, $O(n)$ iterations. Total: $\tilde{O}(mn)$

Recursive Eigenvector: The Worst Case



$\Omega(n)$ nearly-disconnected components

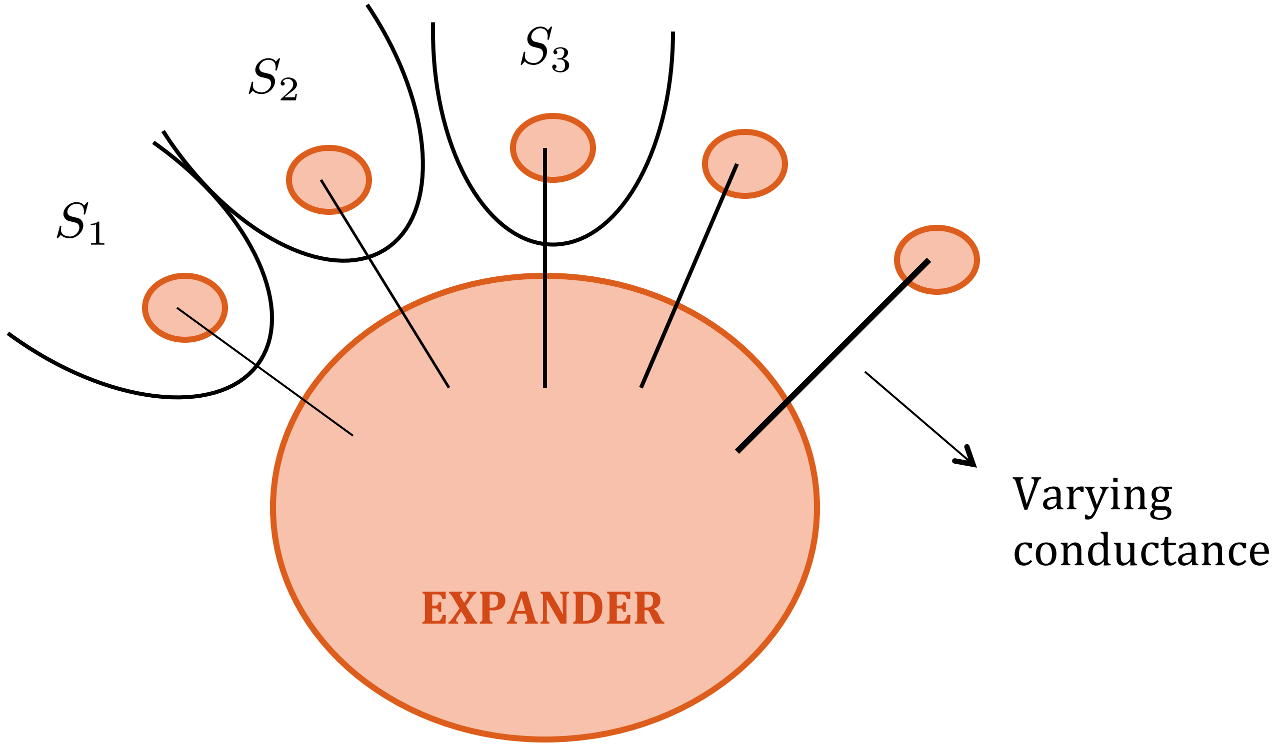
Recursive Eigenvector: The Worst Case



NB: Recursive Eigenvector eliminates one component per iteration.

$\Omega(n)$ iterations are necessary. Each iteration requires $\Omega(m)$ time.

Recursive Eigenvector: The Worst Case

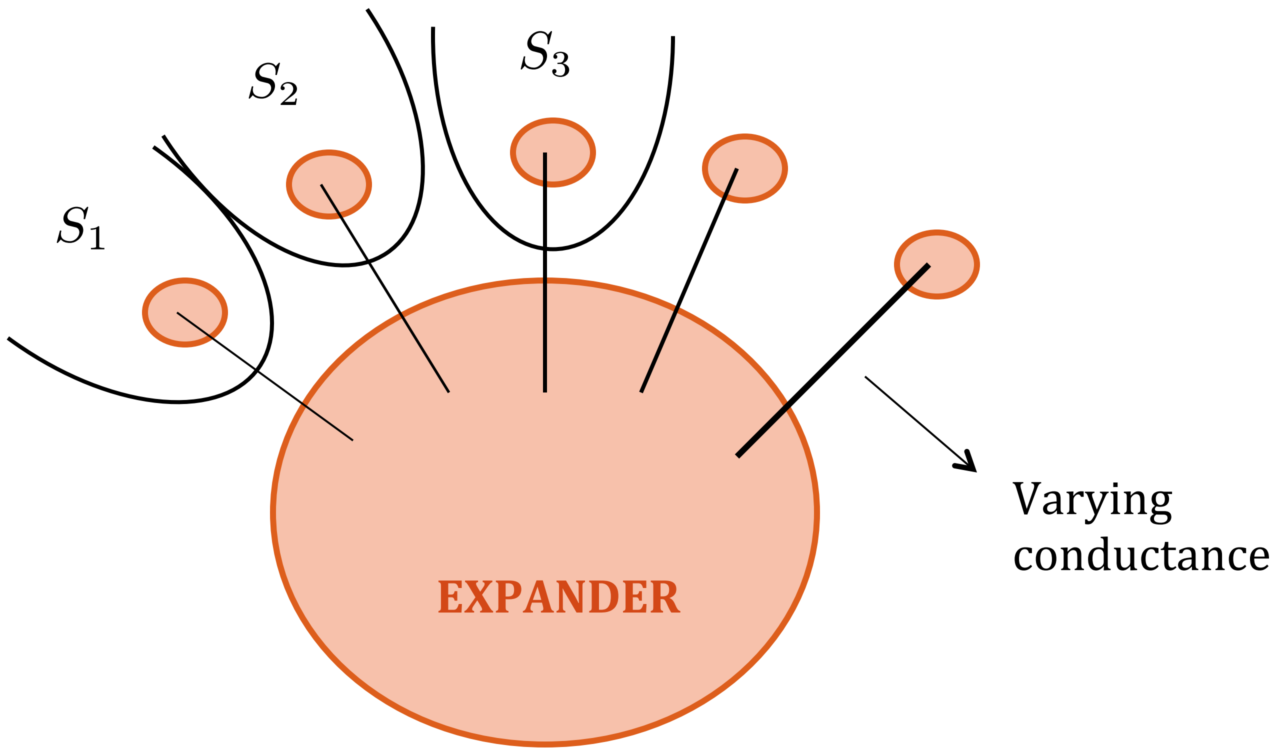


NB: Recursive Eigenvector eliminates one component per iteration.

$\Omega(n)$ iterations are necessary. Each iteration requires $\Omega(mn)$ time.

GOAL: Eliminate unbalanced low-conductance cuts faster.

Recursive Eigenvector: The Worst Case



STABILITY VIEW:

- Ideally, we would like to enforce progress: $\lambda_2(G_{t+1}) >> \lambda_2(G_t)$
- Eigenvector may change completely at every iteration. Impossible to enforce any non-trivial relation between $\lambda_2(G_{t+1})$ and $\lambda_2(G_t)$

Our Algorithm: Contributions

Algorithm	Method	Distinguishes $\geq \gamma$ and	Time
Recursive Eigenvector	Eigenvector	$O(\sqrt{\gamma})$	$\tilde{O}(mn)$
OUR ALGORITHM	Random Walks	$O(\sqrt{\gamma})$	$\tilde{O}(m)$

MAIN FEATURES:

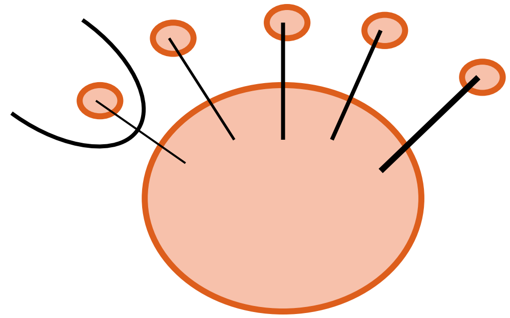
- Compute $O(\log n)$ global heat-kernel random-walk vectors at each iteration
- Unbalanced cuts are removed in $O(\log n)$ iterations
- Method to compute heat-kernel vectors in nearly-linear time

TECHNICAL COMPONENTS:

1) New iterative algorithm with a simple random walk interpretation
2) Novel analysis of Lanczos methods for computing heat-kernel vectors

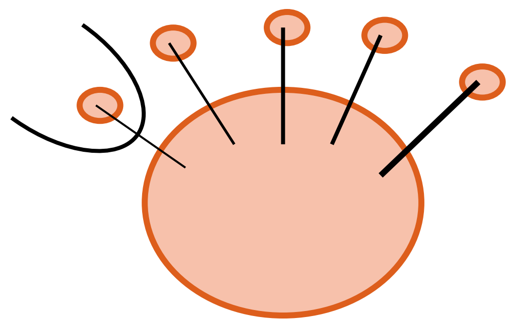
Eliminating Unbalanced Cuts

- The graph eigenvector may be correlated with only one sparse unbalanced cut.

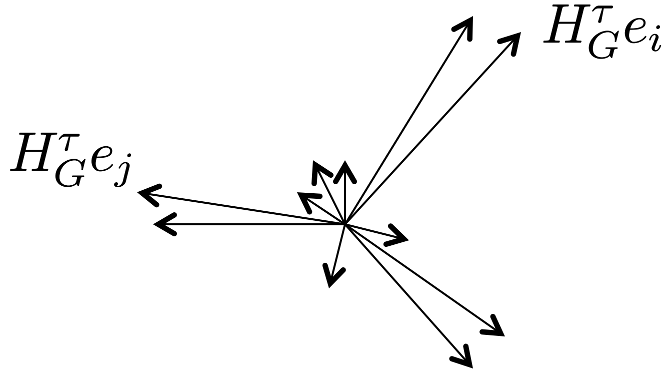


Eliminating Unbalanced Cuts

- The graph eigenvector may be correlated with only one sparse unbalanced cut.



- Consider the Heat-Kernel random walk-matrix H_G^τ for $\tau = \log n/\gamma$.

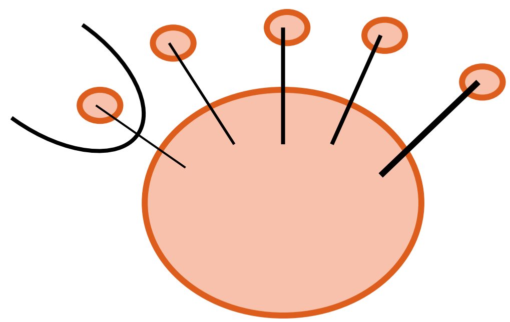


Probability vector for random walk started at vertex i

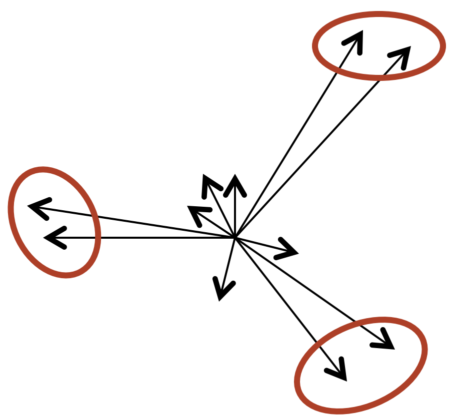
Long vectors are slow-mixing random walks

Eliminating Unbalanced Cuts

- The graph eigenvector may be correlated with only one sparse unbalanced cut.



- Consider the Heat-Kernel random walk-matrix H_G^{τ} for $\tau = \log n/\gamma$.

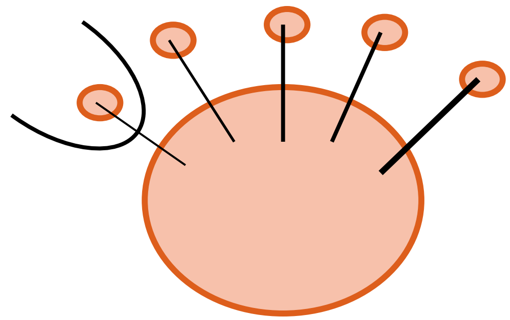


Unbalanced cuts of conductance $< \sqrt{\gamma}$

Eliminating Unbalanced Cuts

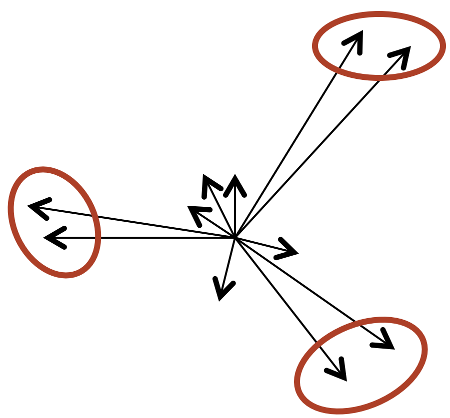
- The graph eigenvector may be correlated with only one sparse unbalanced cut.

SINGLE VECTOR
SINGLE CUT



- Consider the Heat-Kernel random walk-matrix H_G^τ for $\tau = \log n/\gamma$.

VECTOR
EMBEDDING
MULTIPLE CUTS

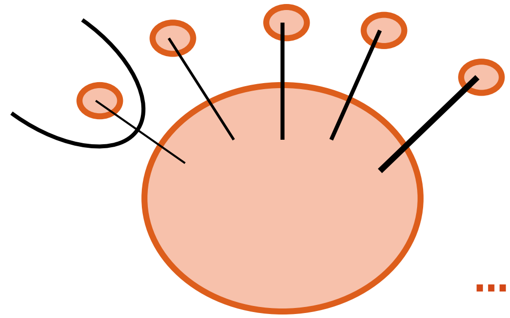


Unbalanced cuts of conductance $< \sqrt{\gamma}$

Eliminating Unbalanced Cuts

- The graph eigenvector may be correlated with only one sparse unbalanced cut.

SINGLE VECTOR
SINGLE CUT

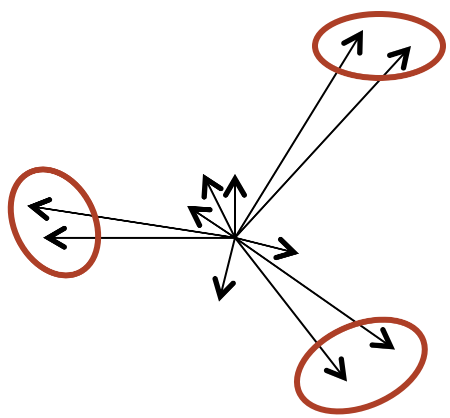


AFTER CUT REMOVAL ...

... eigenvector can change completely

- Consider the Heat-Kernel random walk-matrix H_G^{τ} for $\tau = \log n/\gamma$.

VECTOR
EMBEDDING
MULTIPLE CUTS



... vectors do not change a lot

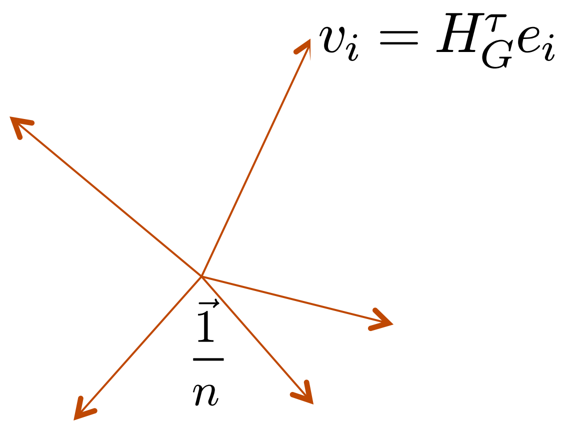
Our Algorithm for Balanced Cut

IDEA BEHIND OUR ALGORITHM:

Replace eigenvector in recursive eigenvector algorithm with

Heat-Kernel random walk H_G^τ for $\tau = \log n/\gamma$

Consider the embedding $\{v_i\}$ given by H_G^τ :



Our Algorithm for Balanced Cut

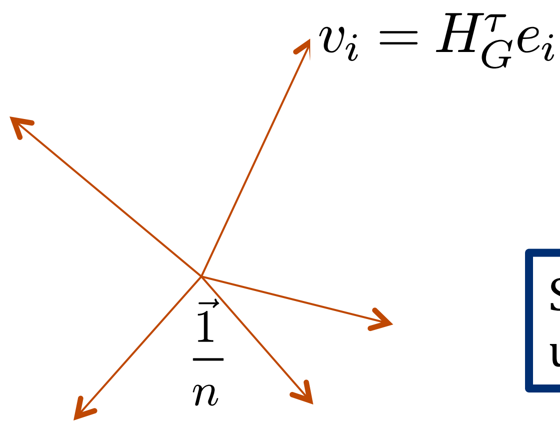
IDEA BEHIND OUR ALGORITHM:

Replace eigenvector in recursive eigenvector algorithm with

Heat-Kernel random walk H_G^τ for $\tau = \log n/\gamma$

Chosen to emphasize cuts of conductance $\approx \gamma$

Consider the embedding $\{v_i\}$ given by H_G^τ :



Stationary distribution is uniform as G is regular

Our Algorithm for Balanced Cut

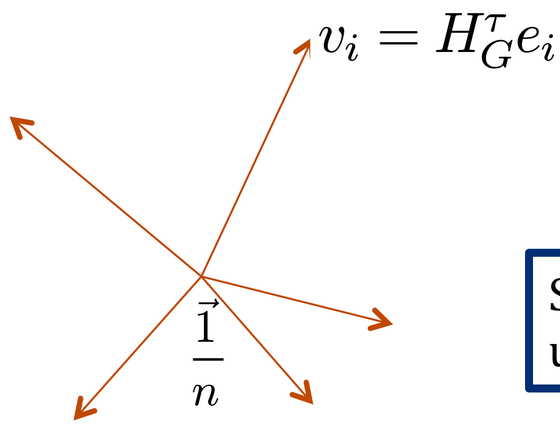
IDEA BEHIND OUR ALGORITHM:

Replace eigenvector in recursive eigenvector algorithm with

Heat-Kernel random walk H_G^τ for $\tau = \log n/\gamma$

Chosen to emphasize cuts of conductance ≈ γ

Consider the embedding $\{v_i\}$ given by H_G^τ :



Stationary distribution is uniform as G is regular

MIXING:

Define the total deviation from stationary for a set $S \subseteq V$ for walk

$$\Psi(H_G^\tau, S) = \sum_{i \in S} ||v_i - \vec{1}/n||^2$$

FUNDAMENTAL QUANTITY TO UNDERSTAND CUTS IN G

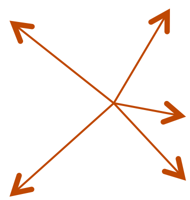
Our Algorithm: Case Analysis

Recall:

$$\tau = \log n/\gamma \qquad \Psi(H_G^\tau, S) = \sum_{i\in S} ||H_G^\tau e_i - \vec{1}/n||^2$$

CASE 1: Random walks have **mixed**

$$v_i = H_G^\tau e_i$$



ALL VECTORS ARE SHORT

$$\Psi(H_G^\tau, V) \leq \frac{1}{\mathrm{poly}(n)}$$

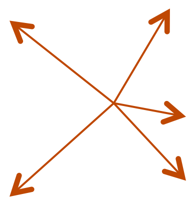
Our Algorithm: Case Analysis

Recall:

$$\tau = \log n/\gamma \qquad \Psi(H_G^\tau, S) = \sum_{i\in S} ||H_G^\tau e_i - \vec{1}/n||^2$$

CASE 1: Random walks have **mixed**

$v_i = H_G^\tau e_i$



ALL VECTORS ARE SHORT

$$\Psi(H_G^\tau, V) \leq \frac{1}{\mathrm{poly}(n)}$$

By definition of τ

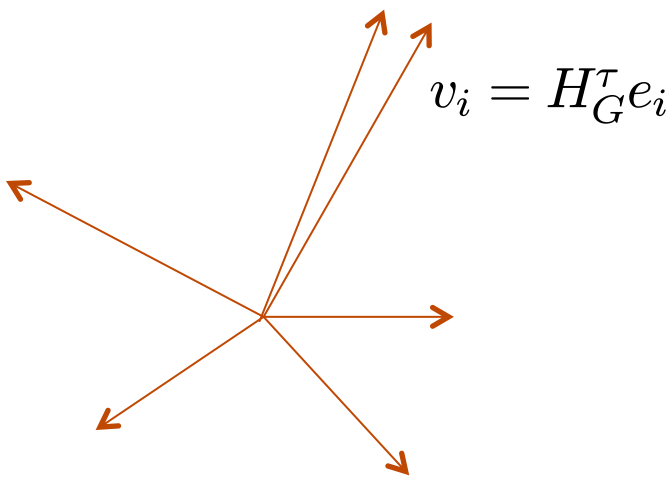
$$\lambda_2 \geq \Omega(\gamma)$$

$$\phi_G \geq \Omega(\gamma)$$

Our Algorithm

$\tau = \log n/\gamma$

$\Psi(H_G^\tau, S) = \sum_{i \in S} ||H_G^\tau e_i - \vec{1}/n||^2$



CASE 2: Random walks have **not mixed**

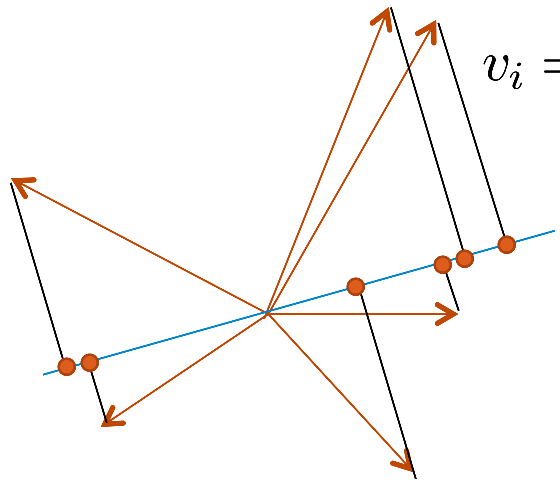
$$\Psi(H_G^\tau, V) > \frac{1}{\text{poly}(n)}$$

We can either find an $\Omega(b)$-balanced cut with conductance $O(\sqrt{\gamma})$

Our Algorithm

$\tau = \log n/\gamma$

$\Psi(H_G^\tau, S) = \sum_{i \in S} ||H_G^\tau e_i - \vec{1}/n||^2$



$v_i = H_G^\tau e_i$

RANDOM PROJECTION
+
SWEEP CUT

CASE 2: Random walks have **not mixed**

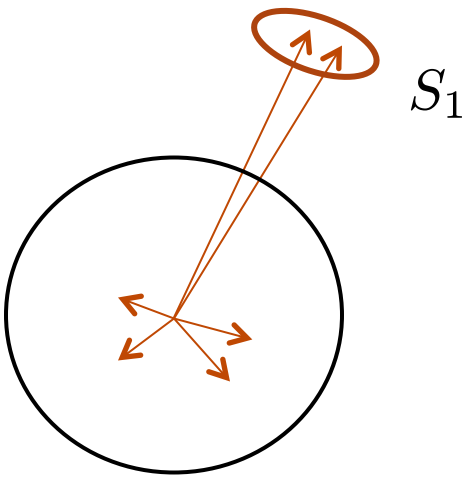
$$\Psi(H_G^\tau, V) > \frac{1}{\text{poly}(n)}$$

We can either find an $\Omega(b)$-balanced cut with conductance $O(\sqrt{\gamma})$

Our Algorithm

$\tau = \log n/\gamma$

$\Psi(H_G^\tau, S) = \sum_{i \in S} ||H_G^\tau e_i - \vec{1}/n||^2$



BALL ROUNDING

CASE 2: Random walks have **not mixed**

$$\Psi(H_G^\tau, V) > \frac{1}{\text{poly}(n)}$$

We can either find an $\Omega(b)$-balanced cut with conductance $O(\sqrt{\gamma})$

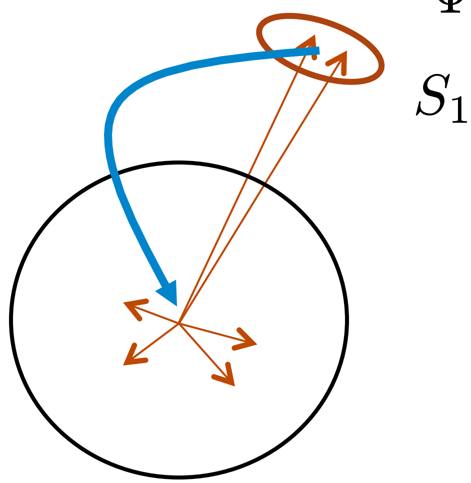
OR a ball cut yields S_1 such that $\phi(S_1) \leq O(\sqrt{\gamma})$ and

$$\boxed{\Psi(H_G^\tau, S_1) \geq \frac{1}{2}\Psi(H_G^\tau, V).}$$

Our Algorithm: Iteration

$\tau = \log n/\gamma$

$\Psi(H_G^\tau, S) = \sum_{i \in S} ||H_G^\tau e_i - \vec{1}/n||^2$



CASE 2: We found an unbalanced cut S_1 with $\phi(S_1) \leq O(\sqrt{\gamma})$ and

$$\boxed{\Psi(H_G^\tau, S_1) \geq \frac{1}{2}\Psi(H_G^\tau, V).}$$

Modify G $=G^{(1)}$ by **adding edges** across $(S_1, \bar{S}_1)$ to construct $G^{(2)}$.

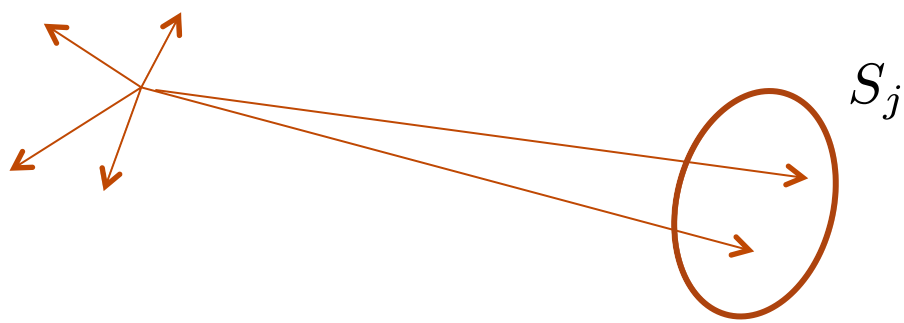
Analogous to removing unbalanced cut S_1
in Recursive Eigenvector algorithm

Our Algorithm: Modifying G

CASE 2: We found an unbalanced cut S_1 with $\phi(S_1) \leq O(\sqrt{\gamma})$ and

$$\Psi(H_G^\tau, S_1) \geq \frac{1}{2}\Psi(H_G^\tau, V).$$

Modify G $=G^{(1)}$ by **adding edges** across $(S_1, \bar{S_1})$ to construct $G^{(2)}$.

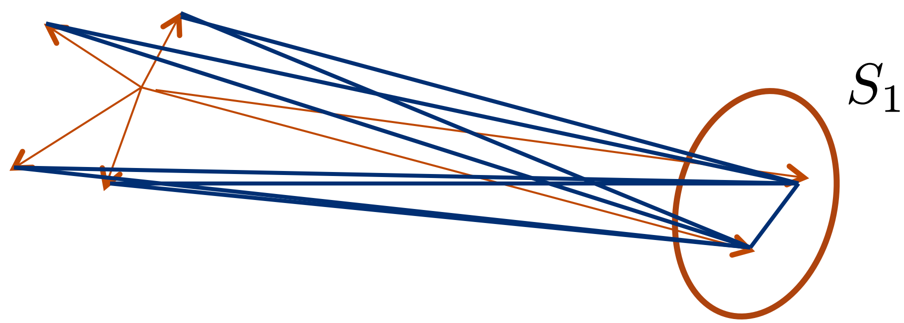


Our Algorithm: Modifying G

CASE 2: We found an unbalanced cut S_1 with $\phi(S_1) \leq O(\sqrt{\gamma})$ and

$$\Psi(H_G^\tau, S_1) \geq \frac{1}{2}\Psi(H_G^\tau, V).$$

Modify G =$G^{(1)}$ by **adding edges** across $(S_1, \bar{S_1})$ to construct $G^{(2)}$.



$$G^{(t+1)} = G^{(t)} + \gamma \sum_{i \in S_t} \mathrm{Star}_i$$

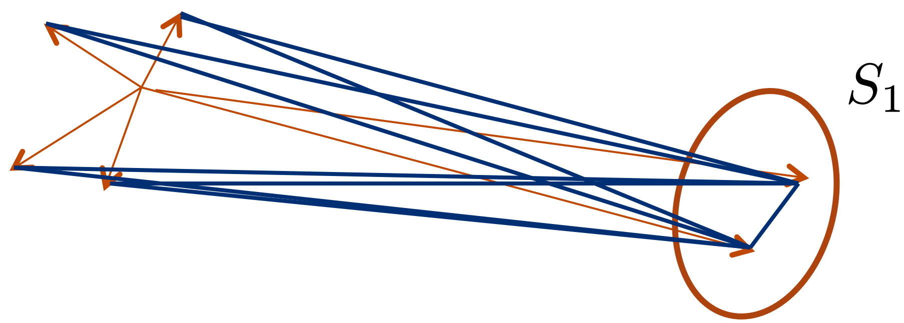
where Star_i is the star graph rooted at vertex i.

Our Algorithm: Modifying G

CASE 2: We found an unbalanced cut S_1 with $\phi(S_1) \leq O(\sqrt{\gamma})$ and

$$\Psi(H_G^\tau, S_1) \geq \frac{1}{2}\Psi(H_G^\tau, V).$$

Modify G = $G^{(1)}$ by **adding edges** across $(S_1, \bar{S_1})$ to construct $G^{(2)}$.



$$G^{(t+1)} = G^{(t)} + \gamma \sum_{i \in S_t} \mathrm{Star}_i$$

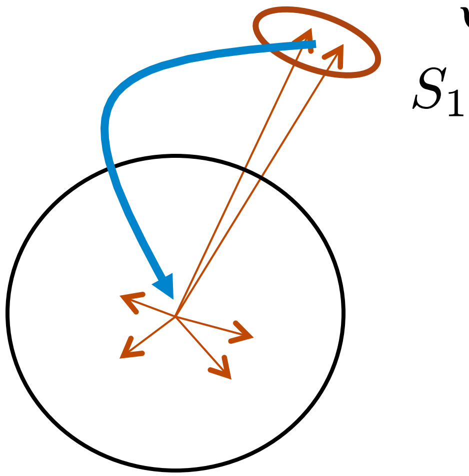
where Star_i is the star graph rooted at vertex i.

The random walk can now escape S_1 more easily.

Our Algorithm: Iteration

$\tau = \log n / \gamma$

$\Psi(H_G^\tau, S) = \sum_{i \in S} ||H_G^\tau e_i - \vec{1}/n||^2$



CASE 2: We found an unbalanced cut S_1 with $\phi(S_1) \leq O(\sqrt{\gamma})$ and

$$\boxed{\Psi(H_G^\tau, S_1) \geq \tfrac{1}{2}\Psi(H_G^\tau, V).}$$

Modify $G = G^{(1)}$ by **adding edges** across $(S_1, \bar{S}_1)$ to construct $G^{(2)}$.

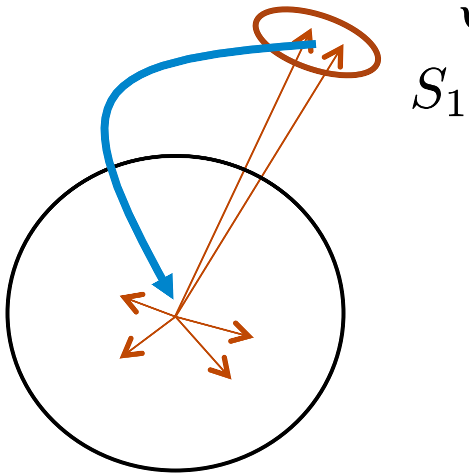
POTENTIAL REDUCTION:

$$\Psi(H^\tau_{G^{(t+1)}}, V) \leq \Psi(H^\tau_{G^{(t)}}, V) - \frac{1}{2}\Psi(H^\tau_{G^{(t)}}, S_t) \leq \frac{3}{4}\Psi(H^\tau_{G^{(t)}}, V)$$

Our Algorithm: Iteration

$\tau = \log n/\gamma$

$\Psi(H_G^\tau, S) = \sum_{i \in S} ||H_G^\tau e_i - \vec{1}/n||^2$



CASE 2: We found an unbalanced cut S_1 with $\phi(S_1) \leq O(\sqrt{\gamma})$ and

$$\boxed{\Psi(H_G^\tau, S_1) \geq \frac{1}{2}\Psi(H_G^\tau, V).}$$

Modify $G = G^{(1)}$ by **adding edges** across $(S_1, \bar{S_1})$ to construct $G^{(2)}$.

POTENTIAL REDUCTION:

$$\Psi(H^\tau_{G^{(t+1)}}, V) \leq \Psi(H^\tau_{G^{(t)}}, V) - \frac{1}{2}\Psi(H^\tau_{G^{(t)}}, S_t) \leq \frac{3}{4}\Psi(H^\tau_{G^{(t)}}, V)$$

CRUCIAL APPLICATION OF STABILITY OF RANDOM WALK

Summary and Potential Analysis

IN SUMMARY:

At every step t of the recursion, we either

1. Produce a $\Omega(b)$-balanced cut of the required conductance, OR

Potential Reduction

IN SUMMARY:

At every step t of the recursion, we either

1. Produce a $\Omega(b)$-balanced cut of the required conductance, OR
2. Find that

$$\Psi(H^{\tau}_{G^{(t)}}, V) \leq \frac{1}{\mathrm{poly}(n)}$$, OR

Potential Reduction

IN SUMMARY:

At every step t of the recursion, we either

1. Produce a $\Omega(b)$-balanced cut of the required conductance, OR
2. Find that

$$\Psi(H^{\tau}_{G^{(t)}}, V) \leq \frac{1}{\mathrm{poly}(n)} \text{, OR}$$

3. Find an unbalanced cut S_t of the required conductance, such that for the graph $G^{(t+1)}$, modified to have increased edges from S_t,

$$\Psi(H^{\tau}_{G^{(t+1)}}, V) \leq \frac{3}{4} \Psi(H^{\tau}_{G^{(t)}}, V)$$

Potential Reduction

IN SUMMARY:

At every step t-1 of the recursion, we either

1. Produce a $\Omega(b)$-balanced cut of the required conductance, OR
2. Find that

$$\Psi(H^{\tau}_{G(t)}, V) \leq \frac{1}{\mathrm{poly}(n)}, \text{ OR}$$

3. Find an unbalanced cut S_t of the required conductance, such that for the process $P^{(t+1)}$, modified to have increased transitions from S_t,

$$\Psi(H^{\tau}_{G(t+1)}, V) \leq \frac{3}{4}\Psi(H^{\tau}_{G(t)}, V)$$

After $T = O(\log n)$ iterations, if no balanced cut is found:

$$\Psi(H^{\tau}_{G(T)}, V) \leq \frac{1}{\mathrm{poly}(n)}$$

From this guarantee, using the definition of $G^{(T)}$, we derive an SDP-certificate that no b-balanced cut of conductance $O(\gamma)$ exists in G.

NB: Only O(log n) iterations to remove unbalanced cuts.

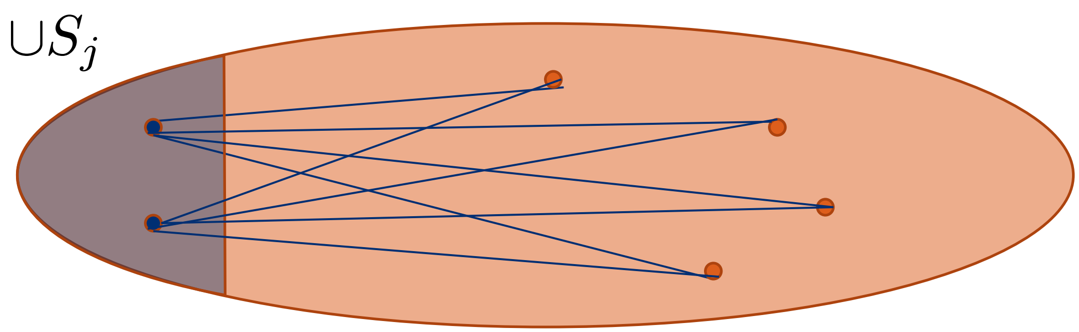
Heat-Kernel and Certificates

- If no balanced cut of conductance is found, our potential analysis yields:

$$\Psi(H^{\tau}_{G^{(T)}}, V) \leq \frac{1}{\mathrm{poly}(n)} \longrightarrow L + \gamma \sum_{j=1}^{T-1} \sum_{i \in S_j} L(\mathrm{Star}_i) \succeq \gamma L(K_V)$$

Modified graph has $\lambda_2 \geq \gamma$

CLAIM: This is a certificate that no balanced cut of conductance < γ existed in G.



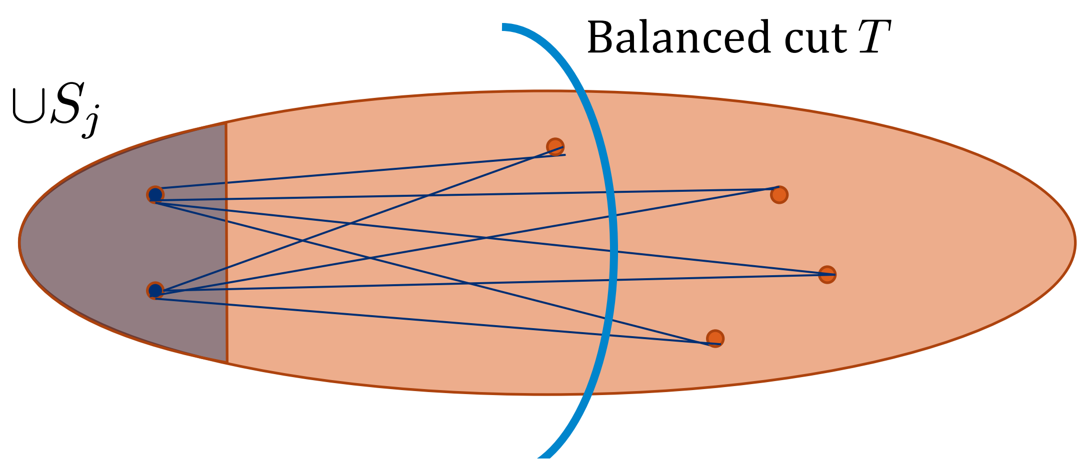
Heat-Kernel and Certificates

- If no balanced cut of conductance is found, our potential analysis yields:

$$\Psi(H^{\tau}_{G(T)}, V) \leq \frac{1}{\mathrm{poly}(n)} \longrightarrow L + \gamma \sum_{j=1}^{T-1} \sum_{i \in S_j} L(\mathrm{Star}_i) \succeq \gamma L(K_V)$$

Modified graph has $\lambda_2 \geq \gamma$

CLAIM: This is a certificate that no balanced cut of conductance < γ existed in G.



$$\phi(T) \geq \gamma - \gamma \frac{|\cup S_j|}{|T|}$$

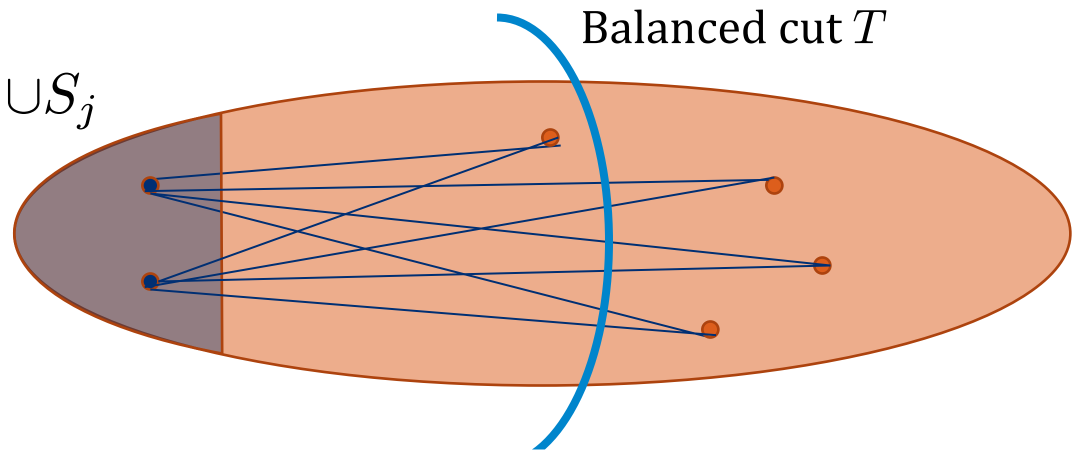
Heat-Kernel and Certificates

- If no balanced cut of conductance is found, our potential analysis yields:

$$\Psi(H^{\tau}_{G^{(T)}}, V) \leq \frac{1}{\text{poly}(n)} \longrightarrow L + \gamma \sum_{j=1}^{T-1} \sum_{i \in S_j} L(\text{Star}_i) \succeq \gamma L(K_V)$$

Modified graph has $\lambda_2 \geq \gamma$

CLAIM: This is a certificate that no balanced cut of conductance < γ existed in G.



$$\phi(T) \geq \gamma - \gamma \frac{|\cup S_j|}{|T|} \geq \gamma - \gamma \frac{b/2}{b} \geq \gamma/2$$

Comparison with Recursive Eigenvector

RECURSIVE EIGENVECTOR:

We can only bound number of iterations by volume of graph removed.

$\Omega(n)$ iterations possible.

OUR ALGORITHM:

Use variance of random walk as potential.

Only $O(\log n)$ iterations necessary.

$$\Psi(P,V) = \underbrace{\textstyle\sum_{i\in V} ||Pe_i - \vec{1}/n||^2}_{\text{STABLE SPECTRAL NOTION OF POTENTIAL}}$$

Running Time

- Our Algorithm runs in $O(\log n)$ iterations.

- In one iteration, we perform some simple computation (projection, sweep cut) on the vector embedding $H^{\tau}_{G(t)}$. This takes time $\tilde{O}(md)$, where d is the dimension of the embedding.

- Can use Johnson-Lindenstrauss to obtain $d = O(\log n)$.

- Hence, we only need to compute $O(\log^2 n)$ matrix-vector products

$$H^{\tau}_{G(t)} u$$

- We show how to perform one such product in time $\tilde{O}(m)$ for all τ.

- OBSTACLE:

  τ, the mean number of steps in the Heat-Kernel random walk, is $\Omega(n^2)$ for path.

Conclusion

<u>NOVEL ALGORITHMIC CONTRIBUTIONS</u>

- Balanced-Cut Algorithm using Random Walks in time $\tilde{O}(m)$

<u>MAIN IDEA</u>

Random walks provide a very useful
stable analogue of the graph eigenvector
via regularization

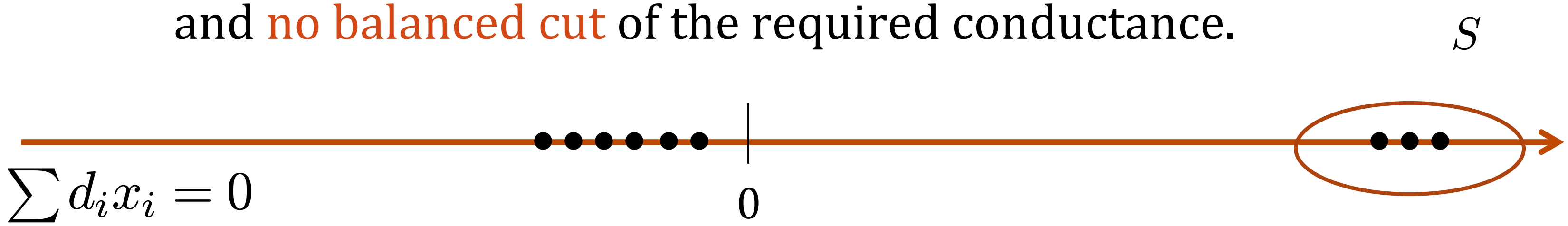
<u>OPEN QUESTION</u>

More applications of this idea?
Applications beyond design of fast algorithms?

A Different Interpretation

THEOREM:

Suppose eigenvector x yields an unbalanced cut S of low conductance and no balanced cut of the required conductance.



Then,

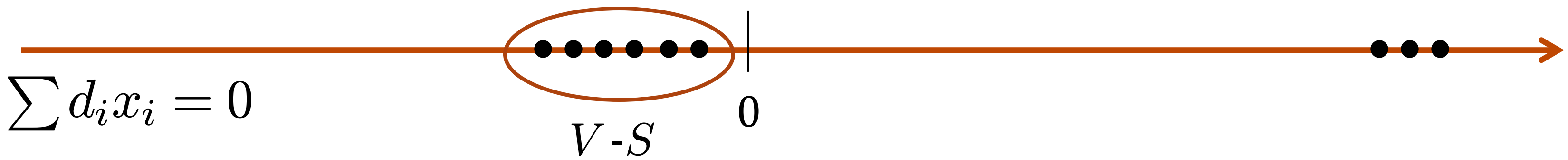
$$\sum_{i \in S} d_i x_i^2 \geq \frac{1}{2} \sum_{i \in V} d_i x_i^2 .$$

In words, S contains most of the variance of the eigenvector.

A Different Interpretation

THEOREM:

Suppose eigenvector x yields an unbalanced cut S of low conductance and no balanced cut of the required conductance.



Then,

$$\sum_{i \in S} d_i x_i^2 \geq \frac{1}{2} \sum_{i \in V} d_i x_i^2.$$

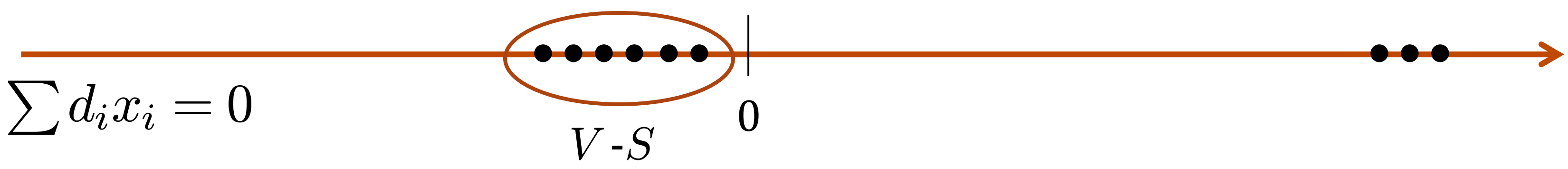
In words, S contains most of the variance of the eigenvector.

QUESTION: Does this mean the graph induced by G on $V - S$ is much closer to have conductance at least γ?

A Different Interpretation

THEOREM:

Suppose eigenvector x yields an unbalanced cut S of low conductance and no balanced cut of the required conductance.



Then,

$$\sum_{i \in S} d_i x_i^2 \geq \frac{1}{2} \sum_{i \in V} d_i x_i^2.$$

QUESTION: Does this mean the graph induced by G on $V-S$ is much closer to have conductance at least γ?

ANSWER: **NO**. x may contain little or no information about G on $V-S$.

Next eigenvector may be only infinitesimally larger.

CONCLUSION: To make significant progress, we need an analogue of the eigenvector that captures sparse

Theorems for Our Algorithm

THEOREM 1: (WALKS HAVE NOT MIXED)

$\Psi(P^{(t)}, V) > \frac{1}{\text{poly}(n)}$ $\longrightarrow$ Can find cut of conductance $O(\sqrt{\gamma})$

Theorems for Our Algorithm

THEOREM 1: (WALKS HAVE NOT MIXED)

$\Psi(P^{(t)}, V) > \frac{1}{\text{poly}(n)}$ $\longrightarrow$ Can find cut of conductance $O(\sqrt{\gamma})$

Proof: Recall that

$P^{(t)} = e^{-\tau Q^{(t)}}$ $\qquad \tau = \log n / \gamma$ $\qquad \Psi(P, V) = \sum_{i \in V} ||Pe_i - \vec{1}/n||^2$

Use the definition of τ. The spectrum of $P^{(t)}$ implies that

$$\sum_{ij \in E} \underbrace{||P^{(t)}e_i - P^{(t)}e_j||^2}_{\text{EDGE LENGTH}} \cdot \quad O(\gamma) \cdot \underbrace{\Psi(P^{(t)}, V)}_{\text{TOTAL VARIANCE}}$$

Theorems for Our Algorithm

THEOREM 1: (WALKS HAVE NOT MIXED)

$$\Psi(P^{(t)}, V) > \frac{1}{\mathrm{poly}(n)} \longrightarrow \text{Can find cut of conductance } O(\sqrt{\gamma})$$

Proof: Recall that

$P^{(t)} = e^{-\tau Q^{(t)}}$ $\quad\quad$ $\tau = \log n/\gamma$ $\quad\quad$ $\Psi(P,V) = \sum_{i \in V} ||Pe_i - \vec{1}/n||^2$

Use the definition of τ. The spectrum of $P^{(t)}$ implies that

$$\sum_{ij \in E} \underbrace{||P^{(t)}e_i - P^{(t)}e_j||^2}_{\text{EDGE LENGTH}} \cdot \quad O(\gamma) \cdot \underbrace{\Psi(P^{(t)}, V)}_{\text{TOTAL VARIANCE}}$$

Hence, by a random projection of the embedding $\{P\, e_i\}$, followed by a sweep cut, we can recover the required cut.

SDP ROUNDING TECHNIQUE

Theorems for Our Algorithm

THEOREM 2: (WALKS HAVE MIXED)

$\Psi(P^{(t)}, V) \cdot \quad \frac{1}{\text{poly}(n)}$ $\longrightarrow$ No $\Omega(b)$-balanced cut of conductance $O(\gamma)$

Theorems for Our Algorithm

THEOREM 2: (WALKS HAVE MIXED)

$\Psi(P^{(t)}, V) \cdot \frac{1}{\text{poly}(n)}$ $\longrightarrow$ No $\Omega(b)$-balanced cut of conductance $O(\gamma)$

Proof: Consider $S = \cup S_i$. Notice that S is unbalanced.

Assumption is equivalent to

$$L(K_V) \bullet e^{-\tau L - O(\log n) \sum_{i \in S} L(S_i)} \cdot \frac{1}{\text{poly}(n)}.$$

Theorems for Our Algorithm

THEOREM 2: (WALKS HAVE MIXED)

$\Psi(P^{(t)}, V) \cdot \frac{1}{\text{poly}(n)}$ $\longrightarrow$ No $\Omega(b)$-balanced cut of conductance $O(\gamma)$

Proof: Consider $S = \cup S_i$. Notice that S is unbalanced.

Assumption is equivalent to

$$L(K_V) \bullet e^{-\tau L - O(\log n) \sum_{i \in S} L(S_i)} \cdot \frac{1}{\text{poly}(n)}.$$

By taking logs,

$$L + O(\gamma) \sum_{i \in S} L(S_i) \succeq \Omega(\gamma) L(K_V).$$

SDP DUAL CERTIFICATE

Theorems for Our Algorithm

THEOREM 2: (WALKS HAVE MIXED)

$\Psi(P^{(t)}, V) \cdot \frac{1}{\text{poly}(n)}$ $\longrightarrow$ No $\Omega(b)$-balanced cut of conductance $O(\gamma)$

Proof: Consider $S = \cup S_i$. Notice that S is unbalanced.

Assumption is equivalent to

$$L(K_V) \bullet e^{-\tau L - O(\log n) \sum_{i \in S} L(S_i)} \cdot \frac{1}{\text{poly}(n)}.$$

By taking logs,

$$L + O(\gamma) \sum_{i \in S} L(S_i) \succeq \Omega(\gamma) L(K_V).$$

SDP DUAL CERTIFICATE

This is a certificate that no $\Omega(1)$-balanced cut of conductance $O(\gamma)$ exists, as evaluating the quadratic form for a vector representing a balanced cut U yields

$$\phi(U) \geq \Omega(\gamma) - \frac{\text{vol}(S)}{\text{vol}(U)} O(\gamma) \geq \Omega(\gamma)$$

as long as S is sufficiently unbalanced.

SDP Interpretation

$$\mathsf{E}_{\{i,j\}\in E_G} \quad ||v_i - v_j||^2 \cdot \quad \gamma,$$

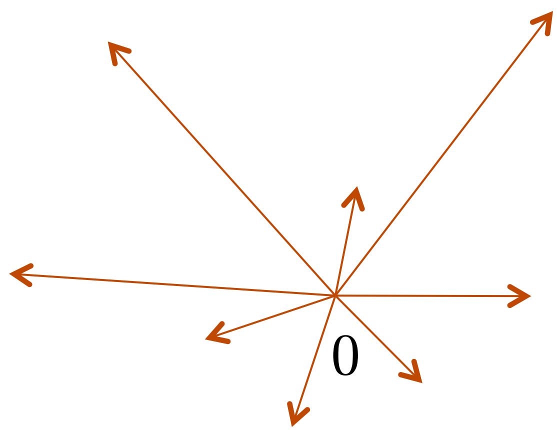
SHORT EDGES

$$\mathsf{E}_{\{i,j\}\in V\times V} \quad ||v_i - v_j||^2 = \frac{1}{2m},$$

FIXED VARIANCE

$$\forall i \in V \qquad \mathsf{E}_{j\in V} \quad ||v_i - v_j||^2 \cdot \quad \frac{1}{b} \cdot \frac{1}{2m}.$$

LENGTH OF STAR EDGES



SDP Interpretation

$$\mathsf{E}_{\{i,j\} \in E_G} \quad ||v_i - v_j||^2 \cdot \quad \gamma,$$

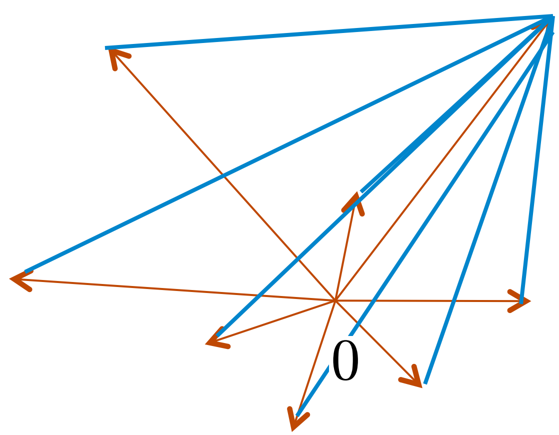
SHORT EDGES

$$\mathsf{E}_{\{i,j\} \in V \times V} \quad ||v_i - v_j||^2 = \frac{1}{2m},$$

FIXED VARIANCE

$$\forall i \in V \qquad \mathsf{E}_{j \in V} \quad ||v_i - v_j||^2 \cdot \quad \frac{1}{b} \cdot \frac{1}{2m}.$$

LENGTH OF STAR EDGES



SHORT RADIUS

Background: Heat-Kernel Random Walk

For simplicity, take G to be **d-regular**.

• The Heat-Kernel Random Walk is a Continuous-Time Markov Chain over V, modeling the diffusion of heat along the edges of G.

• Transitions take place in continuous time t, with an exponential distribution.

$$\frac{\partial p(t)}{\partial t} = -L\frac{p(t)}{d}$$

**Notatio
n**

$$p(t) = e^{-\frac{t}{d}L}p(0) =: \boxed{H_G^t}\; p(0)$$

• The Heat Kernel can be interpreted as Poisson distribution over number of steps of the natural random walk $W = AD^{-1}$:

$$e^{-\frac{t}{d}L} = e^{-t}\sum_{k=1}^{\infty}\frac{t^k}{k!}W^k$$

• In practice, can replace Heat-Kernel with natural random walk W^t