Spectra of the graph.

$M = A/d$ adjacency matrix, A

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: a vector v where $Mv = \lambda v$

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: a vector v where $Mv = \lambda v$

Real, symmetric.

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: a vector v where $Mv = \lambda v$

Real, symmetric.

Claim:
Two eigenvectors with different eigenvalues are orthogonal.

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: a vector v where $Mv = \lambda v$

Real, symmetric.

Claim:
Two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: a vector v where $Mv = \lambda v$

Real, symmetric.

Claim:
Two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T M v' = v^T (\lambda' v')$

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: a vector v where $Mv = \lambda v$

Real, symmetric.

Claim:
Two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T Mv' = v^T(\lambda' v') = \lambda' v^T v'$

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: a vector v where $Mv = \lambda v$

Real, symmetric.

Claim:
Two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T Mv' = v^T(\lambda' v') = \lambda' v^T v'$

$v^T Mv' = \lambda v^T v'$

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: a vector v where $Mv = \lambda v$

Real, symmetric.

Claim:
Two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T M v' = v^T (\lambda' v') = \lambda' v^T v'$

$v^T M v' = \lambda v^T v' = \lambda v^T v.$ □

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: a vector v where $Mv = \lambda v$

Real, symmetric.

Claim:
Two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T Mv' = v^T(\lambda' v') = \lambda' v^T v'$

$v^T Mv' = \lambda v^T v' = \lambda v^T v.$ □

Distinct eigenvalues

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: a vector v where $Mv = \lambda v$

Real, symmetric.

Claim:
Two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T Mv' = v^T(\lambda' v') = \lambda' v^T v'$

$v^T Mv' = \lambda v^T v' = \lambda v^T v.$ $\square$

Distinct eigenvalues $\rightarrow$ orthonormal basis.

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: a vector v where $Mv = \lambda v$

Real, symmetric.

Claim:
Two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T M v' = v^T(\lambda' v') = \lambda' v^T v'$

$v^T M v' = \lambda v^T v' = \lambda v^T v$. □

Distinct eigenvalues $\rightarrow$ orthonormal basis.

In basis: matrix is diagonal..

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: a vector v where $Mv = \lambda v$

Real, symmetric.

Claim:
Two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T Mv' = v^T(\lambda' v') = \lambda' v^T v'$

$v^T Mv' = \lambda v^T v' = \lambda v^T v.$ □

Distinct eigenvalues $\rightarrow$ orthonormal basis.

In basis: matrix is diagonal..

$$M = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

Action of M.

v - assigns weights to vertices.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Action of M.

v - assigns weights to vertices.

  Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value?

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i'$

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v'_i = (M\mathbf{1})_i$

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i' = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i' = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v'_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v'_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v'_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Connected $\rightarrow$ path from x valued node to lower value.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v'_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $v_j < x$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

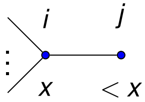
Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i' = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $v_j < x$.



Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

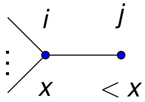
Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v'_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $v_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j)$

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

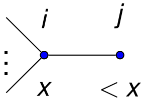
Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v'_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $v_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

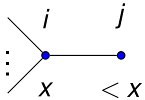
Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i' = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $v_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

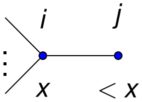
Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v'_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $v_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. □

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

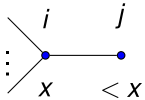
Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i' = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $v_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. □

Claim: Connected if $\lambda_2 < 1$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

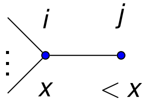
Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v'_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\, e = (i,j)$, $v_i = x$, $v_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. $\square$

Claim: Connected if $\lambda_2 < 1$.
Proof: Assign $+1$ to vertices in one component, $-\delta$ to rest.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

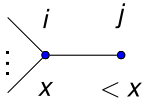
Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v'_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $v_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. □

Claim: Connected if $\lambda_2 < 1$.

Proof: Assign $+1$ to vertices in one component, $-\delta$ to rest.
$x_i = (Mx_i)$

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

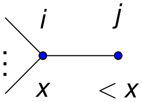
Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i' = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $v_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. $\square$

Claim: Connected if $\lambda_2 < 1$.
Proof: Assign $+1$ to vertices in one component, $-\delta$ to rest.
$x_i = (Mx_i) \implies$ eigenvector with $\lambda = 1$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

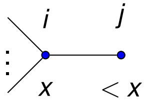
Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v'_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $v_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. $\square$

Claim: Connected if $\lambda_2 < 1$.
Proof: Assign $+1$ to vertices in one component, $-\delta$ to rest.
$x_i = (Mx_i) \implies$ eigenvector with $\lambda = 1$.
Choose δ to make $\sum_i x_i = 0$,

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

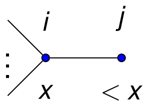
Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v'_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $v_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. $\square$

Claim: Connected if $\lambda_2 < 1$.
Proof: Assign $+1$ to vertices in one component, $-\delta$ to rest.
$x_i = (Mx_i) \implies$ eigenvector with $\lambda = 1$.
Choose δ to make $\sum_i x_i = 0$, i.e., $x \perp \mathbf{1}$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

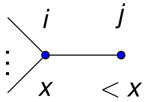
Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v'_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.

Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $v_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. □

Claim: Connected if $\lambda_2 < 1$.
Proof: Assign $+1$ to vertices in one component, $-\delta$ to rest.
$x_i = (Mx_i) \implies$ eigenvector with $\lambda = 1$.
Choose δ to make $\sum_i x_i = 0$, i.e., $x \perp \mathbf{1}$. □

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T A x}{d|x|^2}$ is proxy for "eigenvalue".

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T A x}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T A x$.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T A x}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T A x$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T A v_i}{|v_i|^2} = \lambda_i$.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T Av_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^TAx}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^TAx$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^TAv_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T A x}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T A x$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \to \frac{v_i^T A v_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \to \frac{v_i^T Av_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \to \frac{v_i^T Av_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T Av_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Expander: "no good cuts"

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T Av_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Expander: "no good cuts" "all cuts same".

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T A x}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T A x$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T A v_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Expander: "no good cuts" "all cuts same".

$\implies$ every ± 1 vector is about the same!

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T A x}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T A x$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T A v_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Expander: "no good cuts" "all cuts same".

$\implies$ every ± 1 vector is about the same!

Fast convergence of iterative method.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T Av_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Expander: "no good cuts" "all cuts same".

$\implies$ every ± 1 vector is about the same!

Fast convergence of iterative method.

Intuitively: fast communication across graph,

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T Av_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Expander: "no good cuts" "all cuts same".

$\implies$ every ± 1 vector is about the same!

Fast convergence of iterative method.

Intuitively: fast communication across graph,

$\implies$ fast iterative algorithm.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T Av_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Expander: "no good cuts" "all cuts same".

$\implies$ every ± 1 vector is about the same!

Fast convergence of iterative method.

Intuitively: fast communication across graph,

$\implies$ fast iterative algorithm.

Path (or Cycle)

Consider $\frac{x^T A x}{d|x|^2}$.

Path (or Cycle)

Consider $\frac{x^T A x}{d|x|^2}$.

Random ± 1, same argument as for expander.

Path (or Cycle)

Consider $\frac{x^T A x}{d|x|^2}$.

Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

- Random ± 1, same argument as for expander.
  - $+1/-1$ edges change mass a lot.
  - $\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

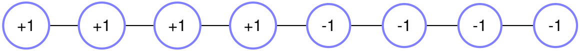
Lose mass only in middle.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

Random ± 1, same argument as for expander.
$+1/-1$ edges change mass a lot.
$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.
Lose mass only in middle.



$\implies \frac{x^T Ax}{d|x|^2} \approx 1 - \frac{1}{n}$.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

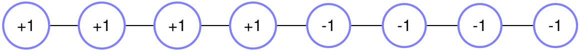
Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Lose mass only in middle.



$\implies \frac{x^T Ax}{d|x|^2} \approx 1 - \frac{1}{n}$.

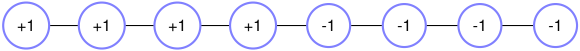
$\implies \lambda_1$ and λ_n differ by a factor of n.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

Random ± 1, same argument as for expander.
$+1/-1$ edges change mass a lot.
$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.
Lose mass only in middle.



$\implies \frac{x^T Ax}{d|x|^2} \approx 1 - \frac{1}{n}$.

$\implies \lambda_1$ and λ_n differ by a factor of n.

Worse example:

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

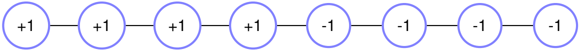
Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Lose mass only in middle.



$\implies \frac{x^T Ax}{d|x|^2} \approx 1 - \frac{1}{n}$.

$\implies \lambda_1$ and λ_n differ by a factor of n.

Worse example:

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

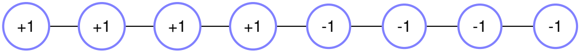
Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Lose mass only in middle.



$\implies \frac{x^T Ax}{d|x|^2} \approx 1 - \frac{1}{n}$.

$\implies \lambda_1$ and λ_n differ by a factor of n.

Worse example:

$\implies \frac{x^T Ax}{|x|^2} \approx 1 - \frac{1}{n^2}$.

Path (or Cycle)

Consider $\frac{x^T A x}{d|x|^2}$.

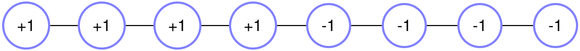
Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T A x}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Lose mass only in middle.



$\implies \frac{x^T A x}{d|x|^2} \approx 1 - \frac{1}{n}$.

$\implies \lambda_1$ and λ_n differ by a factor of n.

Worse example:

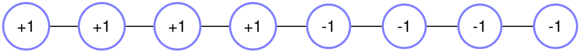
$\implies \frac{x^T A x}{|x|^2} \approx 1 - \frac{1}{n^2}$. For Laplacian $1/n^2$.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

Random ± 1, same argument as for expander.
$+1/-1$ edges change mass a lot.
$\implies \frac{x^T Ax}{d|x|^2} \ll 1$.

Good cut: $+1$ on left half, -1 on right half.
Lose mass only in middle.



$\implies \frac{x^T Ax}{d|x|^2} \approx 1 - \frac{1}{n}$.

$\implies \lambda_1$ and λ_n differ by a factor of n.

Worse example:

$\implies \frac{x^T Ax}{|x|^2} \approx 1 - \frac{1}{n^2}$. For Laplacian $1/n^2$.

Convergence is $O(n^2)$.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

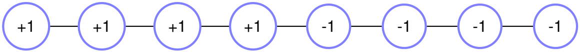
Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Lose mass only in middle.



$\implies \frac{x^T Ax}{d|x|^2} \approx 1 - \frac{1}{n}$.

$\implies \lambda_1$ and λ_n differ by a factor of n.

Worse example:

$\implies \frac{x^T Ax}{|x|^2} \approx 1 - \frac{1}{n^2}$. For Laplacian $1/n^2$.

Convergence is $O(n^2)$. Or $O(n)$.

An aside: Cheeger

Cut with few edges and good “balance”.

An aside: Cheeger

Cut with few edges and good “balance”.

Cut with few edges per unit of vertices “Cut off”.

An aside: Cheeger

Cut with few edges and good “balance”.

Cut with few edges per unit of vertices “Cut off”.

$$h(G) = \min_{S, |S| \leq |V|/2} \frac{E(S, \overline{S})}{d|S|}$$

An aside: Cheeger

Cut with few edges and good “balance”.

Cut with few edges per unit of vertices “Cut off”.

$$h(G) = \min_{S, |S| \leq |V|/2} \frac{E(S, \overline{S})}{d|S|}$$

Expander graph: $h(G) = \Theta(1)$

An aside: Cheeger

Cut with few edges and good “balance”.

Cut with few edges per unit of vertices “Cut off”.

$$h(G) = \min_{S, |S| \leq |V|/2} \frac{E(S, \overline{S})}{d|S|}$$

Expander graph: $h(G) = \Theta(1)$ and $1 - \lambda_2 = \Theta(1)$.

An aside: Cheeger

Cut with few edges and good “balance”.

Cut with few edges per unit of vertices “Cut off”.

$$h(G) = \min_{S, |S| \leq |V|/2} \frac{E(S, \overline{S})}{d|S|}$$

Expander graph: $h(G) = \Theta(1)$ and $1 - \lambda_2 = \Theta(1)$.

Cycle: $h(G) = \Theta(1/n)$.

An aside: Cheeger

Cut with few edges and good “balance”.

Cut with few edges per unit of vertices “Cut off”.

$$h(G) = \min_{S,|S| \le |V|/2} \frac{E(S,\overline{S})}{d|S|}$$

Expander graph: $h(G) = \Theta(1)$ and $1 - \lambda_2 = \Theta(1)$.

Cycle: $h(G) = \Theta(1/n)$. and $1 - \lambda_2 = \Theta(1/n^2)$.

An aside: Cheeger

Cut with few edges and good “balance”.

Cut with few edges per unit of vertices “Cut off”.

$$h(G) = \min_{S, |S| \leq |V|/2} \frac{E(S, \overline{S})}{d|S|}$$

Expander graph: $h(G) = \Theta(1)$ and $1 - \lambda_2 = \Theta(1)$.

Cycle: $h(G) = \Theta(1/n)$. and $1 - \lambda_2 = \Theta(1/n^2)$.

Cheeger’s inequality.

An aside: Cheeger

Cut with few edges and good “balance”.

Cut with few edges per unit of vertices “Cut off”.

$$h(G) = \min_{S, |S| \leq |V|/2} \frac{E(S, \overline{S})}{d|S|}$$

Expander graph: $h(G) = \Theta(1)$ and $1 - \lambda_2 = \Theta(1)$.

Cycle: $h(G) = \Theta(1/n)$. and $1 - \lambda_2 = \Theta(1/n^2)$.

Cheeger’s inequality.

$$\frac{\mu}{2} = \frac{1 - \lambda_2}{2} \leq h(G) \leq \sqrt{2(1 - \lambda_2)} = \sqrt{2\mu}$$

Cycle

Tight example for Other side of Cheeger?

Cycle

Tight example for Other side of Cheeger?

$$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$$

Cycle

Tight example for Other side of Cheeger?

$$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$$

Cycle on n nodes.

Cycle

Tight example for Other side of Cheeger?

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Cycle on n nodes.

Will show other side of Cheeger is tight.

Cycle

Tight example for Other side of Cheeger?

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Cycle on n nodes.

Will show other side of Cheeger is tight.

Edge expansion:Cut in half.

Cycle

Tight example for Other side of Cheeger?

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Cycle on n nodes.

Will show other side of Cheeger is tight.

Edge expansion:Cut in half.
$|S| = n/2$, $|E(S,\overline{S})| = 2$

Cycle

Tight example for Other side of Cheeger?

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Cycle on n nodes.

Will show other side of Cheeger is tight.

Edge expansion:Cut in half.
$|S| = n/2$, $|E(S, \overline{S})| = 2$
$\rightarrow h(G) = \frac{2}{n}$.

Cycle

Tight example for Other side of Cheeger?

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Cycle on n nodes.

Will show other side of Cheeger is tight.

Edge expansion:Cut in half.
$|S| = n/2$, $|E(S,\overline{S})| = 2$
$\rightarrow h(G) = \frac{2}{n}$.

Show eigenvalue gap $\mu \leq \frac{1}{n^2}$.

Cycle

Tight example for Other side of Cheeger?

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Cycle on n nodes.

Will show other side of Cheeger is tight.

Edge expansion:Cut in half.
$|S| = n/2$, $|E(S,\overline{S})| = 2$
$\rightarrow h(G) = \frac{2}{n}$.

Show eigenvalue gap $\mu \leq \frac{1}{n^2}$.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T M x}{x^T x}$ close to 1.

Slow vector.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T M x}{x^T x}$ close to 1.

Slow vector.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T M x}{x^T x}$ close to 1.

$$x_i = \begin{cases} i - n/4 & \text{if } i \leq n/2 \\ 3n/4 - i & \text{if } i > n/2 \end{cases}$$

Slow vector.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T M x}{x^T x}$ close to 1.

$$x_i = \begin{cases} i - n/4 & \text{if } i \leq n/2 \\ 3n/4 - i & \text{if } i > n/2 \end{cases}$$

Hit with M.

Slow vector.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T M x}{x^T x}$ close to 1.

$$x_i = \begin{cases} i - n/4 & \text{if } i \leq n/2 \\ 3n/4 - i & \text{if } i > n/2 \end{cases}$$

Hit with M.

$$(Mx)_i = \begin{cases} -n/4 + 1/2 & \text{if } i = 1, n \\ n/4 - 1 & \text{if } i = n/2 \\ x_i & \text{otherwise} \end{cases}$$

Slow vector.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T Mx}{x^T x}$ close to 1.

$$x_i = \begin{cases} i - n/4 & \text{if } i \leq n/2 \\ 3n/4 - i & \text{if } i > n/2 \end{cases}$$

Hit with M.

$$(Mx)_i = \begin{cases} -n/4 + 1/2 & \text{if } i = 1, n \\ n/4 - 1 & \text{if } i = n/2 \\ x_i & \text{otherwise} \end{cases}$$

$\rightarrow x^T Mx = x^T x(1 - O(\frac{1}{n^2}))$

Slow vector.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T M x}{x^T x}$ close to 1.

$$x_i = \begin{cases} i - n/4 & \text{if } i \leq n/2 \\ 3n/4 - i & \text{if } i > n/2 \end{cases}$$

Hit with M.

$$(Mx)_i = \begin{cases} -n/4 + 1/2 & \text{if } i = 1, n \\ n/4 - 1 & \text{if } i = n/2 \\ x_i & \text{otherwise} \end{cases}$$

$\rightarrow x^T M x = x^T x(1 - O(\frac{1}{n^2})) \quad \rightarrow \lambda_2 \geq 1 - O(\frac{1}{n^2})$

Slow vector.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T Mx}{x^T x}$ close to 1.

$$x_i = \begin{cases} i - n/4 & \text{if } i \leq n/2 \\ 3n/4 - i & \text{if } i > n/2 \end{cases}$$

Hit with M.

$$(Mx)_i = \begin{cases} -n/4 + 1/2 & \text{if } i = 1, n \\ n/4 - 1 & \text{if } i = n/2 \\ x_i & \text{otherwise} \end{cases}$$

$\rightarrow x^T Mx = x^T x(1 - O(\frac{1}{n^2})) \quad \rightarrow \lambda_2 \geq 1 - O(\frac{1}{n^2})$

$\mu = \lambda_1 - \lambda_2 = O(\frac{1}{n^2})$

Slow vector.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T M x}{x^T x}$ close to 1.

$$x_i = \begin{cases} i - n/4 & \text{if } i \leq n/2 \\ 3n/4 - i & \text{if } i > n/2 \end{cases}$$

Hit with M.

$$(Mx)_i = \begin{cases} -n/4 + 1/2 & \text{if } i = 1, n \\ n/4 - 1 & \text{if } i = n/2 \\ x_i & \text{otherwise} \end{cases}$$

$\rightarrow x^T M x = x^T x (1 - O(\frac{1}{n^2})) \quad \rightarrow \lambda_2 \geq 1 - O(\frac{1}{n^2})$

$\mu = \lambda_1 - \lambda_2 = O(\frac{1}{n^2})$

$h(G) = \frac{2}{n} = \Theta(\sqrt{\mu})$

Slow vector.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T M x}{x^T x}$ close to 1.

$$x_i = \begin{cases} i - n/4 & \text{if } i \leq n/2 \\ 3n/4 - i & \text{if } i > n/2 \end{cases}$$

Hit with M.

$$(Mx)_i = \begin{cases} -n/4 + 1/2 & \text{if } i = 1, n \\ n/4 - 1 & \text{if } i = n/2 \\ x_i & \text{otherwise} \end{cases}$$

$\rightarrow x^T M x = x^T x(1 - O(\frac{1}{n^2})) \quad \rightarrow \lambda_2 \geq 1 - O(\frac{1}{n^2})$

$\mu = \lambda_1 - \lambda_2 = O(\frac{1}{n^2})$

$h(G) = \frac{2}{n} = \Theta(\sqrt{\mu})$

$\frac{\mu}{2}$

Slow vector.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T M x}{x^T x}$ close to 1.

$$x_i = \begin{cases} i - n/4 & \text{if } i \leq n/2 \\ 3n/4 - i & \text{if } i > n/2 \end{cases}$$

Hit with M.

$$(Mx)_i = \begin{cases} -n/4 + 1/2 & \text{if } i = 1, n \\ n/4 - 1 & \text{if } i = n/2 \\ x_i & \text{otherwise} \end{cases}$$

$\rightarrow x^T M x = x^T x(1 - O(\frac{1}{n^2})) \quad \rightarrow \lambda_2 \geq 1 - O(\frac{1}{n^2})$

$\mu = \lambda_1 - \lambda_2 = O(\frac{1}{n^2})$

$h(G) = \frac{2}{n} = \Theta(\sqrt{\mu})$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2}$

Slow vector.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T M x}{x^T x}$ close to 1.

$$x_i = \begin{cases} i - n/4 & \text{if } i \leq n/2 \\ 3n/4 - i & \text{if } i > n/2 \end{cases}$$

Hit with M.

$$(Mx)_i = \begin{cases} -n/4 + 1/2 & \text{if } i = 1, n \\ n/4 - 1 & \text{if } i = n/2 \\ x_i & \text{otherwise} \end{cases}$$

$\rightarrow x^T M x = x^T x (1 - O(\frac{1}{n^2})) \quad \rightarrow \lambda_2 \geq 1 - O(\frac{1}{n^2})$

$\mu = \lambda_1 - \lambda_2 = O(\frac{1}{n^2})$

$h(G) = \frac{2}{n} = \Theta(\sqrt{\mu})$

$\frac{\mu}{2} = \frac{1 - \lambda_2}{2} \leq h(G)$

Slow vector.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T Mx}{x^T x}$ close to 1.

$$x_i = \begin{cases} i - n/4 & \text{if } i \leq n/2 \\ 3n/4 - i & \text{if } i > n/2 \end{cases}$$

Hit with M.

$$(Mx)_i = \begin{cases} -n/4 + 1/2 & \text{if } i = 1, n \\ n/4 - 1 & \text{if } i = n/2 \\ x_i & \text{otherwise} \end{cases}$$

$\rightarrow x^T Mx = x^T x(1 - O(\frac{1}{n^2})) \quad \rightarrow \lambda_2 \geq 1 - O(\frac{1}{n^2})$

$\mu = \lambda_1 - \lambda_2 = O(\frac{1}{n^2})$

$h(G) = \frac{2}{n} = \Theta(\sqrt{\mu})$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)}$

Slow vector.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T Mx}{x^T x}$ close to 1.

$$x_i = \begin{cases} i - n/4 & \text{if } i \leq n/2 \\ 3n/4 - i & \text{if } i > n/2 \end{cases}$$

Hit with M.

$$(Mx)_i = \begin{cases} -n/4 + 1/2 & \text{if } i = 1, n \\ n/4 - 1 & \text{if } i = n/2 \\ x_i & \text{otherwise} \end{cases}$$

$\rightarrow x^T Mx = x^T x(1 - O(\frac{1}{n^2})) \quad \rightarrow \lambda_2 \geq 1 - O(\frac{1}{n^2})$

$\mu = \lambda_1 - \lambda_2 = O(\frac{1}{n^2})$

$h(G) = \frac{2}{n} = \Theta(\sqrt{\mu})$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Slow vector.

Find $x \perp \mathbf{1}$ with Rayleigh quotient, $\frac{x^T M x}{x^T x}$ close to 1.

$$x_i = \begin{cases} i - n/4 & \text{if } i \leq n/2 \\ 3n/4 - i & \text{if } i > n/2 \end{cases}$$

Hit with M.

$$(Mx)_i = \begin{cases} -n/4 + 1/2 & \text{if } i = 1, n \\ n/4 - 1 & \text{if } i = n/2 \\ x_i & \text{otherwise} \end{cases}$$

$\rightarrow x^T M x = x^T x (1 - O(\frac{1}{n^2})) \quad \rightarrow \lambda_2 \geq 1 - O(\frac{1}{n^2})$

$\mu = \lambda_1 - \lambda_2 = O(\frac{1}{n^2})$

$h(G) = \frac{2}{n} = \Theta(\sqrt{\mu})$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Tight example for upper bound for Cheeger.

Eigenvalues of cycle?

Eigenvalues: $\cos \frac{2\pi k}{n}$.

Eigenvalues of cycle?

Eigenvalues: $\cos \frac{2\pi k}{n}$.

$x_i = \cos \frac{2\pi k i}{n}$

Eigenvalues of cycle?

Eigenvalues: $\cos \frac{2\pi k}{n}$.

$x_i = \cos \frac{2\pi k i}{n}$

$(Mx)_i$

Eigenvalues of cycle?

Eigenvalues: $\cos\frac{2\pi k}{n}$.

$x_i = \cos\frac{2\pi k i}{n}$

$(Mx)_i = \cos\left(\frac{2\pi k(i+1)}{n}\right) + \cos\left(\frac{2\pi k(i-1)}{n}\right)$

Eigenvalues of cycle?

Eigenvalues: $\cos \frac{2\pi k}{n}$.

$x_i = \cos \frac{2\pi k i}{n}$

$$(Mx)_i = \cos\left(\frac{2\pi k(i+1)}{n}\right) + \cos\left(\frac{2\pi k(i-1)}{n}\right) = 2\cos\left(\frac{2\pi k}{n}\right)\cos\left(\frac{2\pi k i}{n}\right)$$

Eigenvalues of cycle?

Eigenvalues: $\cos\frac{2\pi k}{n}$.

$x_i = \cos\frac{2\pi ki}{n}$

$$(Mx)_i = \cos\left(\frac{2\pi k(i+1)}{n}\right) + \cos\left(\frac{2\pi k(i-1)}{n}\right) = 2\cos\left(\frac{2\pi k}{n}\right)\cos\left(\frac{2\pi ki}{n}\right)$$

Eigenvalue: $\propto \cos\frac{2\pi k}{n}$.

Eigenvalues of cycle?

Eigenvalues: $\cos \frac{2\pi k}{n}$.

$x_i = \cos \frac{2\pi k i}{n}$

$(Mx)_i = \cos\left(\frac{2\pi k(i+1)}{n}\right) + \cos\left(\frac{2\pi k(i-1)}{n}\right) = 2\cos\left(\frac{2\pi k}{n}\right)\cos\left(\frac{2\pi k i}{n}\right)$

Eigenvalue: $\propto \cos \frac{2\pi k}{n}$.

Eigenvalues:

Eigenvalues of cycle?

Eigenvalues: $\cos \frac{2\pi k}{n}$.

$x_i = \cos \frac{2\pi k i}{n}$

$(Mx)_i = \cos\left(\frac{2\pi k(i+1)}{n}\right) + \cos\left(\frac{2\pi k(i-1)}{n}\right) = 2\cos\left(\frac{2\pi k}{n}\right)\cos\left(\frac{2\pi k i}{n}\right)$

Eigenvalue: $\propto \cos \frac{2\pi k}{n}$.

Eigenvalues:
vibration modes of system.

Eigenvalues of cycle?

Eigenvalues: $\cos \frac{2\pi k}{n}$.

$x_i = \cos \frac{2\pi ki}{n}$

$$(Mx)_i = \cos\left(\frac{2\pi k(i+1)}{n}\right) + \cos\left(\frac{2\pi k(i-1)}{n}\right) = 2\cos\left(\frac{2\pi k}{n}\right)\cos\left(\frac{2\pi ki}{n}\right)$$

Eigenvalue: $\propto \cos \frac{2\pi k}{n}$.

Eigenvalues:

vibration modes of system.

Fourier basis.

Eigenvalues of cycle?

Eigenvalues: $\cos\frac{2\pi k}{n}$.

$x_i = \cos\frac{2\pi k i}{n}$

$$(Mx)_i = \cos\left(\frac{2\pi k(i+1)}{n}\right) + \cos\left(\frac{2\pi k(i-1)}{n}\right) = 2\cos\left(\frac{2\pi k}{n}\right)\cos\left(\frac{2\pi k i}{n}\right)$$

Eigenvalue: $\propto \cos\frac{2\pi k}{n}$.

Eigenvalues:
- vibration modes of system.
- Fourier basis.

Hypercube

$V = \{0, 1\}^d$

Hypercube

$V = \{0, 1\}^d \quad (x, y) \in E$

Hypercube

$V = \{0,1\}^d \quad (x,y) \in E$ when x and y differ in one bit.

Hypercube

$V = \{0,1\}^d \quad (x,y) \in E$ when x and y differ in one bit.

$|V| = 2^d$

Hypercube

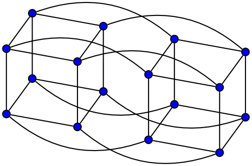
$V = \{0,1\}^d \quad (x,y) \in E$ when x and y differ in one bit.

$|V| = 2^d$ $|E| = d2^{d-1}$.

Hypercube

$V = \{0,1\}^d \quad (x,y) \in E$ when x and y differ in one bit.

$|V| = 2^d \; |E| = d2^{d-1}$.

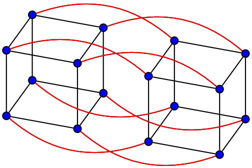


Good cuts?

Hypercube

$V = \{0,1\}^d$ $\quad (x,y) \in E$ when x and y differ in one bit.

$|V| = 2^d$ $|E| = d2^{d-1}$.

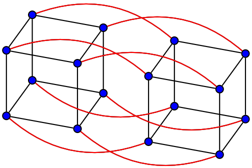


Good cuts? "Coordinate cut": d of them.

Hypercube

$V = \{0,1\}^d \quad (x,y) \in E$ when x and y differ in one bit.

$|V| = 2^d$ $|E| = d2^{d-1}$.



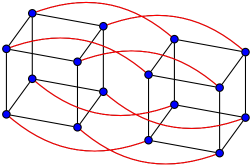
Good cuts? "Coordinate cut": d of them.

Edge expansion:

Hypercube

$V = \{0,1\}^d \quad (x,y) \in E$ when x and y differ in one bit.

$|V| = 2^d$ $|E| = d2^{d-1}$.



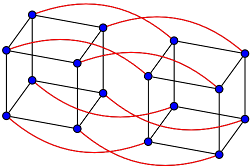
Good cuts? "Coordinate cut": d of them.

Edge expansion: $\frac{2^{d-1}}{d2^{d-1}}$

Hypercube

$V = \{0,1\}^d \quad (x,y) \in E$ when x and y differ in one bit.

$|V| = 2^d$ $|E| = d2^{d-1}$.



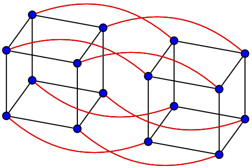
Good cuts? "Coordinate cut": d of them.

Edge expansion: $\frac{2^{d-1}}{d2^{d-1}} = \frac{1}{d}$

Hypercube

$V = \{0,1\}^d \quad (x,y) \in E$ when x and y differ in one bit.

$|V| = 2^d$ $|E| = d2^{d-1}$.



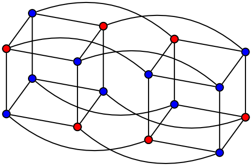
Good cuts? "Coordinate cut": d of them.
Edge expansion: $\frac{2^{d-1}}{d2^{d-1}} = \frac{1}{d}$

Ball cut: All nodes within $d/2$ of node, say $00\cdots0$.

Hypercube

$V = \{0,1\}^d$ $(x,y) \in E$ when x and y differ in one bit.

$|V| = 2^d$ $|E| = d2^{d-1}$.



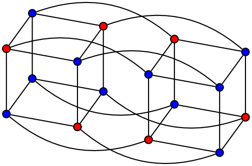
Good cuts? "Coordinate cut": d of them.
Edge expansion: $\frac{2^{d-1}}{d2^{d-1}} = \frac{1}{d}$

Ball cut: All nodes within $d/2$ of node, say $00\cdots0$.
Vertex cut size:

Hypercube

$V = \{0,1\}^d$ $\quad (x,y) \in E$ when x and y differ in one bit.

$|V| = 2^d$ $|E| = d2^{d-1}$.



Good cuts? "Coordinate cut": d of them.

Edge expansion: $\frac{2^{d-1}}{d2^{d-1}} = \frac{1}{d}$

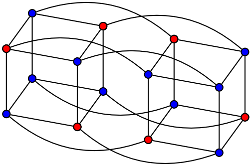
Ball cut: All nodes within $d/2$ of node, say $00\cdots0$.

Vertex cut size: $\binom{d}{d/2}$ bit strings with $d/2$ 1's.

Hypercube

$V = \{0,1\}^d$ $(x,y) \in E$ when x and y differ in one bit.

$|V| = 2^d$ $|E| = d2^{d-1}$.



Good cuts? "Coordinate cut": d of them.

Edge expansion: $\frac{2^{d-1}}{d2^{d-1}} = \frac{1}{d}$

Ball cut: All nodes within $d/2$ of node, say $00\cdots0$.

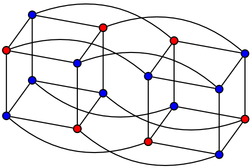
Vertex cut size: $\binom{d}{d/2}$ bit strings with $d/2$ 1's.

$\approx \frac{2^d}{\sqrt{d}}$

Hypercube

$V = \{0,1\}^d$ $\quad (x,y) \in E$ when x and y differ in one bit.

$|V| = 2^d$ $|E| = d2^{d-1}$.



Good cuts? "Coordinate cut": d of them.
Edge expansion: $\frac{2^{d-1}}{d2^{d-1}} = \frac{1}{d}$

Ball cut: All nodes within $d/2$ of node, say $00\cdots0$.
Vertex cut size: $\binom{d}{d/2}$ bit strings with $d/2$ 1's.

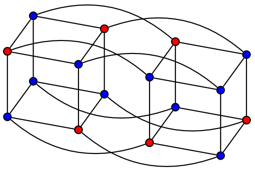
$$\approx \frac{2^d}{\sqrt{d}}$$

Vertex expansion: $\approx \frac{1}{\sqrt{d}}$.

Hypercube

$V = \{0,1\}^d$ $\quad (x,y) \in E$ when x and y differ in one bit.

$|V| = 2^d$ $|E| = d2^{d-1}$.



Good cuts? "Coordinate cut": d of them.

Edge expansion: $\frac{2^{d-1}}{d2^{d-1}} = \frac{1}{d}$

Ball cut: All nodes within $d/2$ of node, say $00\cdots0$.

Vertex cut size: $\binom{d}{d/2}$ bit strings with $d/2$ 1's.

$$\approx \frac{2^d}{\sqrt{d}}$$

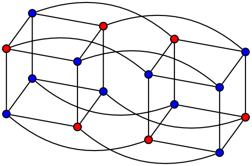
Vertex expansion: $\approx \frac{1}{\sqrt{d}}$.

Edge expansion: $d/2$ edges to next level. $\approx \frac{1}{2\sqrt{d}}$

Hypercube

$V = \{0,1\}^d$ $\quad (x,y) \in E$ when x and y differ in one bit.

$|V| = 2^d$ $|E| = d2^{d-1}$.



Good cuts? "Coordinate cut": d of them.

Edge expansion: $\frac{2^{d-1}}{d2^{d-1}} = \frac{1}{d}$

Ball cut: All nodes within $d/2$ of node, say $00\cdots0$.

Vertex cut size: $\binom{d}{d/2}$ bit strings with $d/2$ 1's.

$$\approx \frac{2^d}{\sqrt{d}}$$

Vertex expansion: $\approx \frac{1}{\sqrt{d}}$.

Edge expansion: $d/2$ edges to next level. $\approx \frac{1}{2\sqrt{d}}$

Worse by a factor of $\sqrt{d}$

Eigenvalues of hypercube.

Anyone see any symmetry?

Eigenvalues of hypercube.

Anyone see any symmetry?

Coordinate cuts. +1 on one side, -1 on other.

Eigenvalues of hypercube.

Anyone see any symmetry?

Coordinate cuts. +1 on one side, -1 on other.

$(Mv)_i = (1-2/d)v_i$.

Eigenvalues of hypercube.

Anyone see any symmetry?

Coordinate cuts. +1 on one side, -1 on other.

$(Mv)_i = (1-2/d)v_i$.

Eigenvalue $1-2/d$.

Eigenvalues of hypercube.

Anyone see any symmetry?

Coordinate cuts. +1 on one side, -1 on other.

$(Mv)_i = (1-2/d)v_i$.

Eigenvalue $1-2/d$. d Eigenvectors.

Eigenvalues of hypercube.

Anyone see any symmetry?

Coordinate cuts. +1 on one side, -1 on other.

$(Mv)_i = (1 - 2/d)v_i$.

Eigenvalue $1 - 2/d$. d Eigenvectors. Why orthogonal?

Eigenvalues of hypercube.

Anyone see any symmetry?

Coordinate cuts. +1 on one side, -1 on other.

$(Mv)_i = (1-2/d)v_i$.

Eigenvalue $1-2/d$. d Eigenvectors. Why orthogonal?

Next eigenvectors?

Eigenvalues of hypercube.

Anyone see any symmetry?

Coordinate cuts. +1 on one side, -1 on other.

$(Mv)_i = (1 - 2/d)v_i$.

Eigenvalue $1 - 2/d$. d Eigenvectors. Why orthogonal?

Next eigenvectors?

Delete edges in two dimensions.

Eigenvalues of hypercube.

Anyone see any symmetry?

Coordinate cuts. +1 on one side, -1 on other.

$(Mv)_i = (1-2/d)v_i$.

Eigenvalue $1-2/d$. d Eigenvectors. Why orthogonal?

Next eigenvectors?

Delete edges in two dimensions.

Four subcubes: bipartite.

Eigenvalues of hypercube.

Anyone see any symmetry?

Coordinate cuts. +1 on one side, -1 on other.

$(Mv)_i = (1-2/d)v_i$.

Eigenvalue $1-2/d$. d Eigenvectors. Why orthogonal?

Next eigenvectors?

Delete edges in two dimensions.

Four subcubes: bipartite. Color ± 1

Eigenvalues of hypercube.

Anyone see any symmetry?

Coordinate cuts. +1 on one side, -1 on other.

$(Mv)_i = (1-2/d)v_i$.

Eigenvalue $1-2/d$. d Eigenvectors. Why orthogonal?

Next eigenvectors?

Delete edges in two dimensions.

Four subcubes: bipartite. Color ± 1

Eigenvalue:

Eigenvalues of hypercube.

Anyone see any symmetry?

Coordinate cuts. +1 on one side, -1 on other.

$(Mv)_i = (1-2/d)v_i$.

Eigenvalue $1-2/d$. d Eigenvectors. Why orthogonal?

Next eigenvectors?

Delete edges in two dimensions.

Four subcubes: bipartite. Color ± 1

Eigenvalue: $1-4/d$.

Eigenvalues of hypercube.

Anyone see any symmetry?

Coordinate cuts. +1 on one side, -1 on other.

$(Mv)_i = (1-2/d)v_i$.

Eigenvalue $1-2/d$. d Eigenvectors. Why orthogonal?

Next eigenvectors?

Delete edges in two dimensions.

Four subcubes: bipartite. Color ± 1

Eigenvalue: $1-4/d$. $\binom{d}{2}$ eigenvectors.

Eigenvalues of hypercube.

Anyone see any symmetry?

Coordinate cuts. +1 on one side, -1 on other.

$(Mv)_i = (1-2/d)v_i$.

Eigenvalue $1-2/d$. d Eigenvectors. Why orthogonal?

Next eigenvectors?

Delete edges in two dimensions.

Four subcubes: bipartite. Color ± 1

Eigenvalue: $1-4/d$. $\binom{d}{2}$ eigenvectors.

Eigenvalues: $1-2k/d$.

Eigenvalues of hypercube.

Anyone see any symmetry?

Coordinate cuts. +1 on one side, -1 on other.

$(Mv)_i = (1-2/d)v_i$.

Eigenvalue $1-2/d$. d Eigenvectors. Why orthogonal?

Next eigenvectors?

Delete edges in two dimensions.

Four subcubes: bipartite. Color ± 1

Eigenvalue: $1-4/d$. $\binom{d}{2}$ eigenvectors.

Eigenvalues: $1-2k/d$. $\binom{d}{k}$ eigenvectors.

Back to Cheeger.

Coordinate Cuts:

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$.

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$\frac{\mu}{2}$

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$\frac{\mu}{2} = \frac{1-\lambda_2}{2}$

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1 - 2/d$. d Eigenvectors.

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G)$

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)}$$

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$$

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

For hypercube: $h(G) = \frac{1}{d}$

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

For hypercube: $h(G) = \frac{1}{d}$ $\lambda_1 - \lambda_2 = 2/d$.
Left hand side is tight.

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$$

For hypercube: $h(G) = \frac{1}{d}$ $\lambda_1 - \lambda_2 = 2/d$.
Left hand side is tight.

Note: hamming weight vector also in first eigenspace.

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$$

For hypercube: $h(G) = \frac{1}{d}$ $\lambda_1 - \lambda_2 = 2/d$.
Left hand side is tight.

Note: hamming weight vector also in first eigenspace.

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

For hypercube: $h(G) = \frac{1}{d}$ $\lambda_1 - \lambda_2 = 2/d$.
Left hand side is tight.

Note: hamming weight vector also in first eigenspace.

Lose “names” in hypercube, find coordinate cut?

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

For hypercube: $h(G) = \frac{1}{d}$ $\lambda_1 - \lambda_2 = 2/d$.
Left hand side is tight.

Note: hamming weight vector also in first eigenspace.

Lose “names” in hypercube, find coordinate cut?

Find coordinate cut?

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

For hypercube: $h(G) = \frac{1}{d}$ $\lambda_1 - \lambda_2 = 2/d$.
Left hand side is tight.

Note: hamming weight vector also in first eigenspace.

Lose "names" in hypercube, find coordinate cut?

Find coordinate cut?

Eigenvector v maps to line.

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

For hypercube: $h(G) = \frac{1}{d}$ $\lambda_1 - \lambda_2 = 2/d$.
Left hand side is tight.

Note: hamming weight vector also in first eigenspace.

Lose "names" in hypercube, find coordinate cut?

Find coordinate cut?

Eigenvector v maps to line.
Cut along line.

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

For hypercube: $h(G) = \frac{1}{d}$ $\lambda_1 - \lambda_2 = 2/d$.
Left hand side is tight.

Note: hamming weight vector also in first eigenspace.

Lose “names” in hypercube, find coordinate cut?

Find coordinate cut?

Eigenvector v maps to line.
Cut along line.

Eigenvector algorithm gets a linear combination of coordinate cuts.

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

For hypercube: $h(G) = \frac{1}{d}$ $\lambda_1 - \lambda_2 = 2/d$.
Left hand side is tight.

Note: hamming weight vector also in first eigenspace.

Lose "names" in hypercube, find coordinate cut?

Find coordinate cut?

Eigenvector v maps to line.
Cut along line.

Eigenvector algorithm gets a linear combination of coordinate cuts.

Something like ball cut.

Back to Cheeger.

Coordinate Cuts:
Eigenvalue $1-2/d$. d Eigenvectors.

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

For hypercube: $h(G) = \frac{1}{d}$ $\lambda_1 - \lambda_2 = 2/d$.
Left hand side is tight.

Note: hamming weight vector also in first eigenspace.

Lose "names" in hypercube, find coordinate cut?

Find coordinate cut?

Eigenvector v maps to line.
Cut along line.

Eigenvector algorithm gets a linear combination of coordinate cuts.

Something like ball cut.

Find coordinate cut?

Random Walk.

p - probability distribution.

Random Walk.

p - probability distribution.

Probability distrubtion after choose a random neighbor.

Random Walk.

p - probability distribution.

Probability distrubtion after choose a random neighbor.
Mp.

Random Walk.

p - probability distribution.

Probability distrubtion after choose a random neighbor.
Mp.

Converge to uniform distribution.

Random Walk.

p - probability distribution.

Probability distrubtion after choose a random neighbor.
Mp.

Converge to uniform distribution.

Power method: $M^t x$ goes to highest eigenvector.

Random Walk.

p - probability distribution.

Probability distrubtion after choose a random neighbor.
Mp.

Converge to uniform distribution.

Power method: $M^t x$ goes to highest eigenvector.

$M^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2 v_2 + \cdots$

Random Walk.

p - probability distribution.

Probability distrubtion after choose a random neighbor.
Mp.

Converge to uniform distribution.

Power method: $M^t x$ goes to highest eigenvector.

$M^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2 v_2 + \cdots$

rate of convergence $\propto \lambda_1 - \lambda_2$

Random Walk.

p - probability distribution.

Probability distrubtion after choose a random neighbor.
Mp.

Converge to uniform distribution.

Power method: $M^t x$ goes to highest eigenvector.

$M^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2 v_2 + \cdots$

rate of convergence $\propto \lambda_1 - \lambda_2$

$\Omega(n^2)$ steps to get close to uniform.

Random Walk.

p - probability distribution.

Probability distrubtion after choose a random neighbor.
Mp.

Converge to uniform distribution.

Power method: $M^t x$ goes to highest eigenvector.

$M^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2 v_2 + \cdots$

rate of convergence $\propto \lambda_1 - \lambda_2$

$\Omega(n^2)$ steps to get close to uniform.

Start at node 0, probability distribution, $[1, 0, 0, \cdots, 0]$.

Random Walk.

p - probability distribution.

Probability distrubtion after choose a random neighbor.
Mp.

Converge to uniform distribution.

Power method: $M^t x$ goes to highest eigenvector.

$M^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2 v_2 + \cdots$

rate of convergence $\propto \lambda_1 - \lambda_2$

$\Omega(n^2)$ steps to get close to uniform.

Start at node 0, probability distribution, $[1, 0, 0, \cdots, 0]$.
Takes $\Omega(n^2)$ to get n steps away.

Random Walk.

p - probability distribution.

Probability distrubtion after choose a random neighbor.
  Mp.

Converge to uniform distribution.

Power method: $M^t x$ goes to highest eigenvector.

$M^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2 v_2 + \cdots$

rate of convergence $\propto \lambda_1 - \lambda_2$

$\Omega(n^2)$ steps to get close to uniform.

Start at node 0, probability distribution, $[1, 0, 0, \cdots, 0]$.
  Takes $\Omega(n^2)$ to get n steps away.

Recall drunken sailor.