The Product Cut

Thomas Laurent

Loyola Marymount University

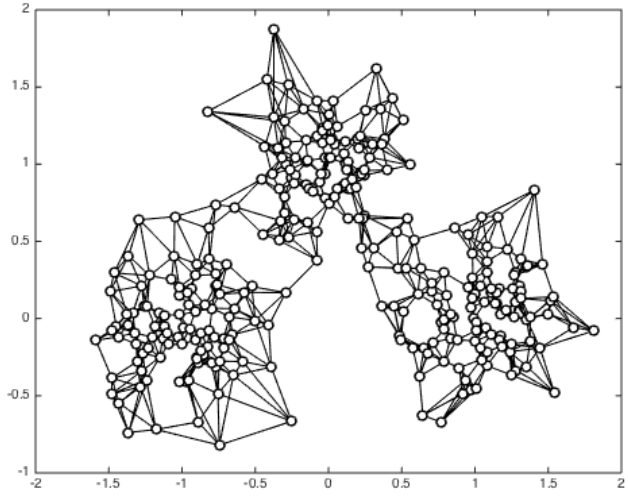
November 27, 2016

Collaborators:

- James von Brecht (Cal. State. Long Beach)
- Xavier Bresson (EPFL)
- Arthur Szlam (Facebook AI)

INPUTS

- A graph:

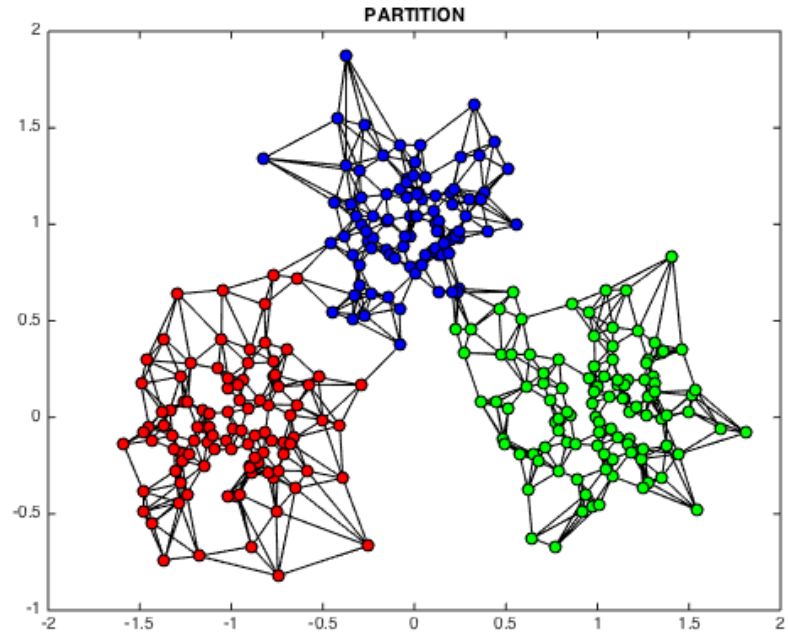


- Number of groups:

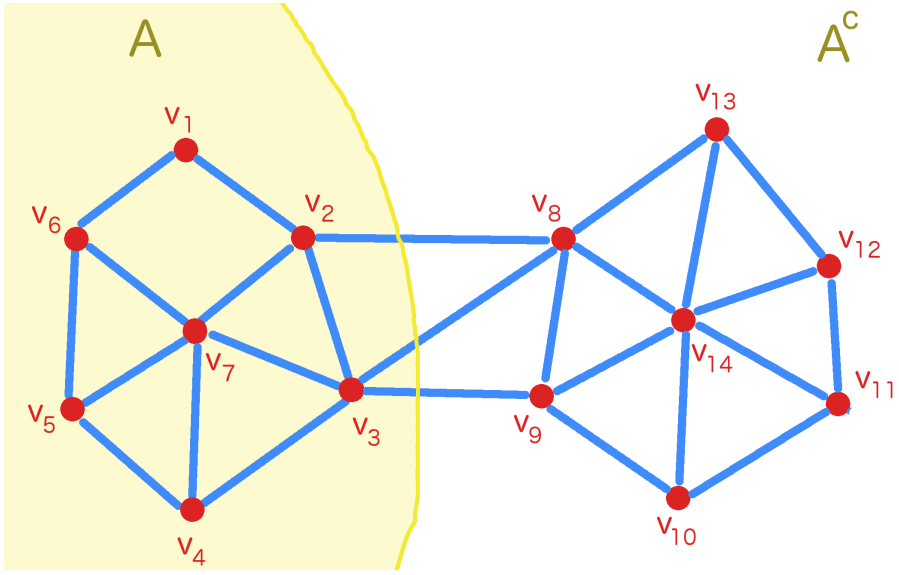
$$\boxed{R = 3}$$

OUTPUT

- A partition of the graph in R groups:



The classical Normalized Cut



$V = \{v_1, \ldots, v_{14}\}$ vertex set

$$w_{ij} = \begin{cases} 1 & \text{if } v_i \text{ connected to } v_j \\ 0 & \text{otherwise} \end{cases}$$

Cut and Conductance of a set $A \subset V$

$$\text{Cut}(A, A^c) = \sum_{i \in A} \sum_{j \in A^c} w_{ij} \qquad \text{Cond}(A) = \frac{\text{Cut}(A, A^c)}{\text{Vol}(A)}$$

where $\text{Vol}(A) = \sum_{i \in A} d_i$

The Conductance

- $\mathrm{Cond}(A) = \dfrac{\mathrm{Cut}(A, A^c)}{\mathrm{Vol}(A)}$

- $\mathrm{Cond}(A) = \dfrac{\text{nb. of edges with one vertex in } A \text{ and one vertex in } A^c}{\text{nb. of edges with one vertex in } A}$

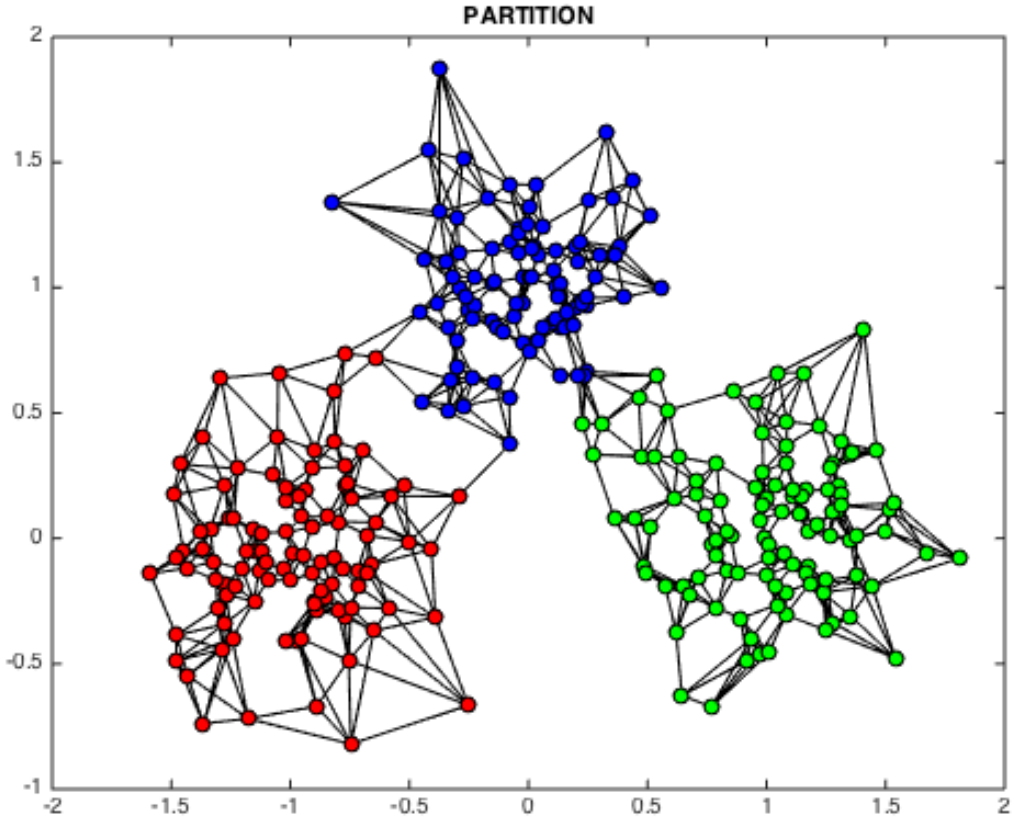
- $\mathrm{Cond}(A) =$ ratio of edges leaving A $\qquad (0 \leq \mathrm{Cond}(A) \leq 1)$

- A is a "cluster" if $\mathrm{Cond}(A) < 1/2$

The Normalized Cut objective

Minimize $$\frac{1}{R}\sum_{r=1}^{R}\frac{\mathrm{Cut}(A_r, A_r^c)}{\mathrm{Vol}(A_r)}$$

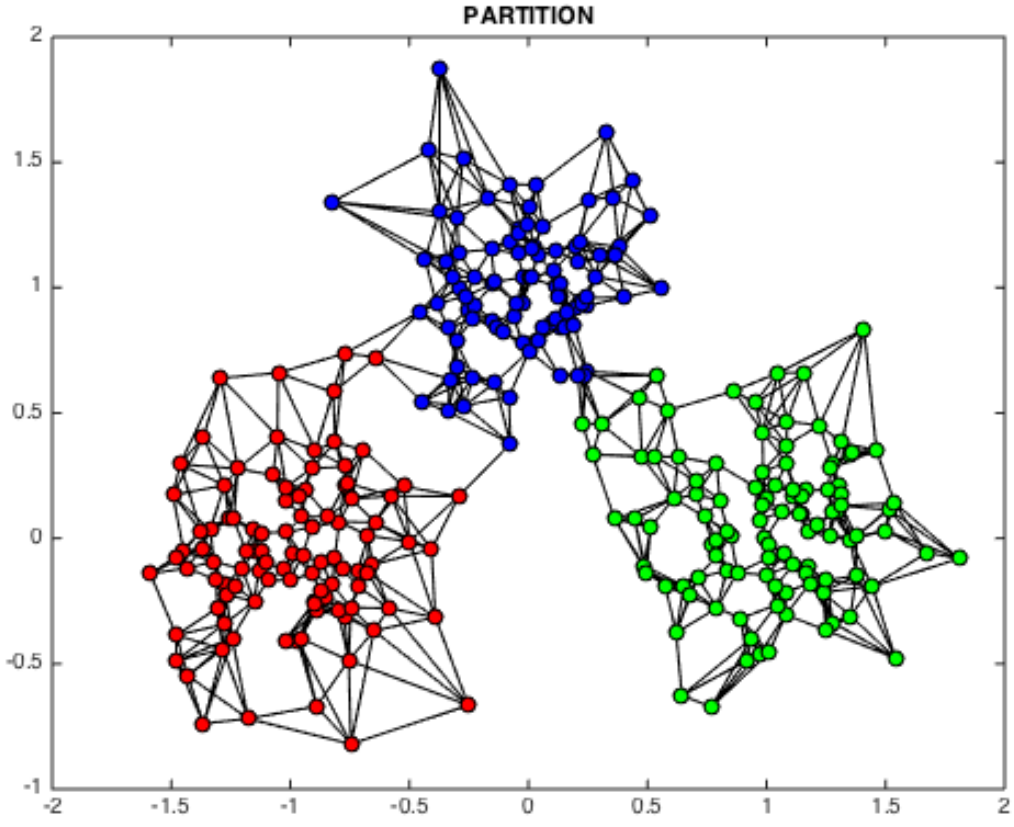
over all partitions $(A_1, \ldots, A_R)$ of the vertex set V



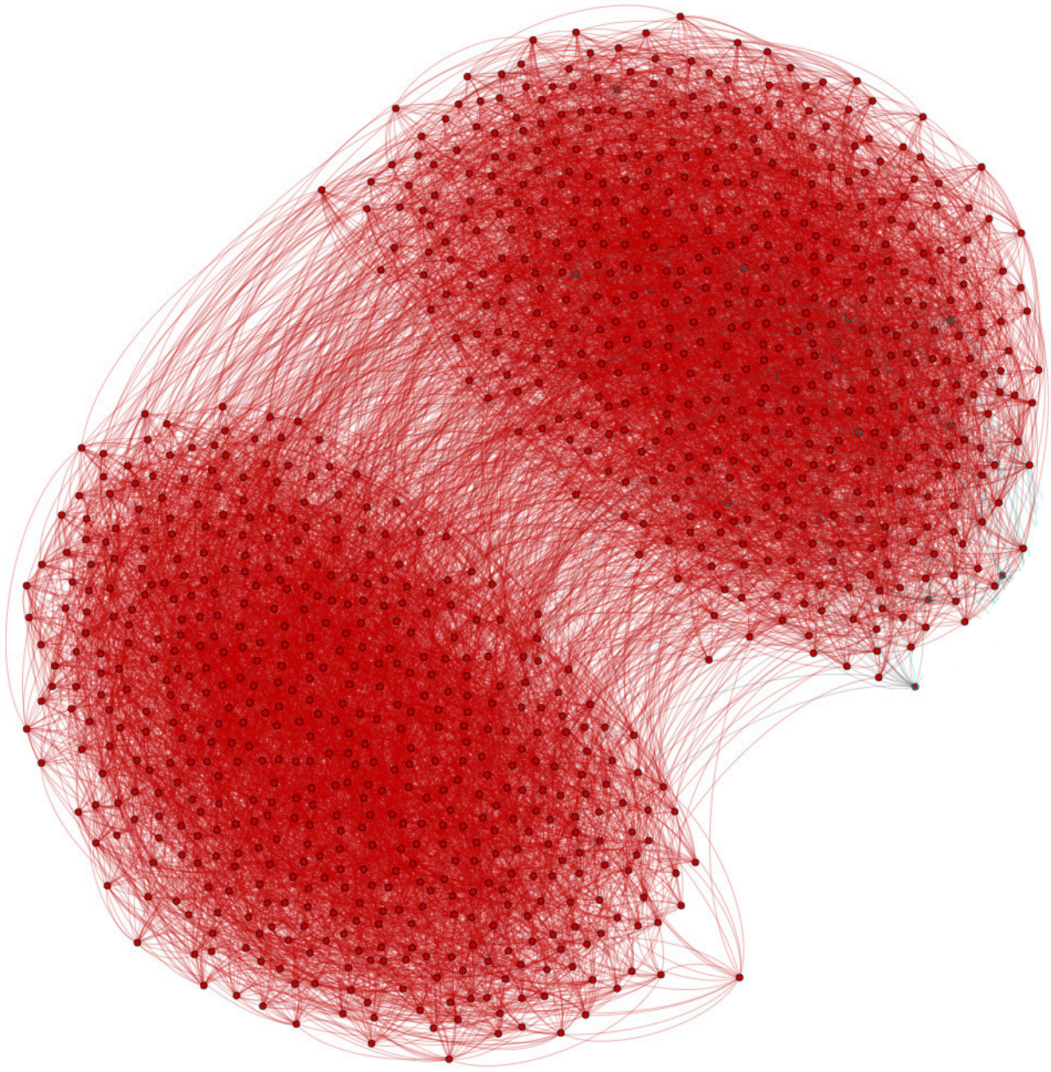
The Cut objective

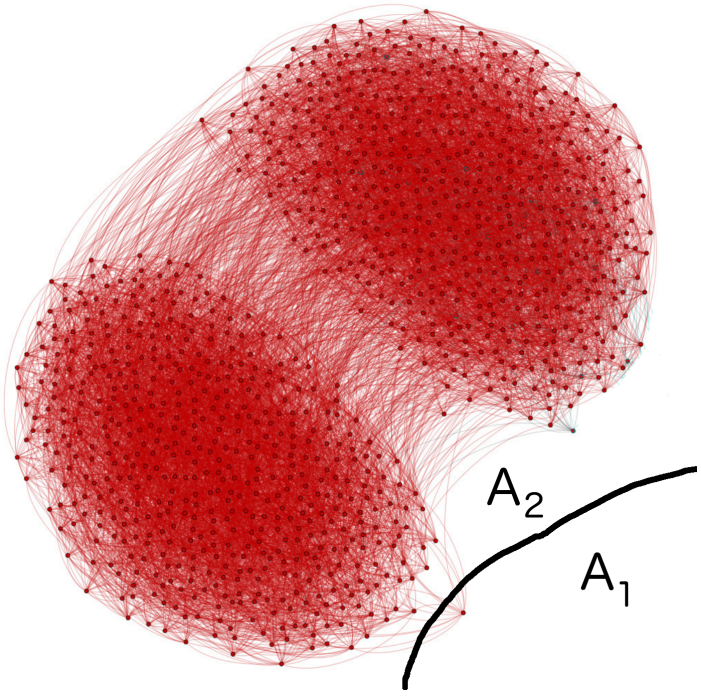
Minimize $$\sum_{r=1}^{R} \mathrm{Cut}(A_r, A_r^c)$$

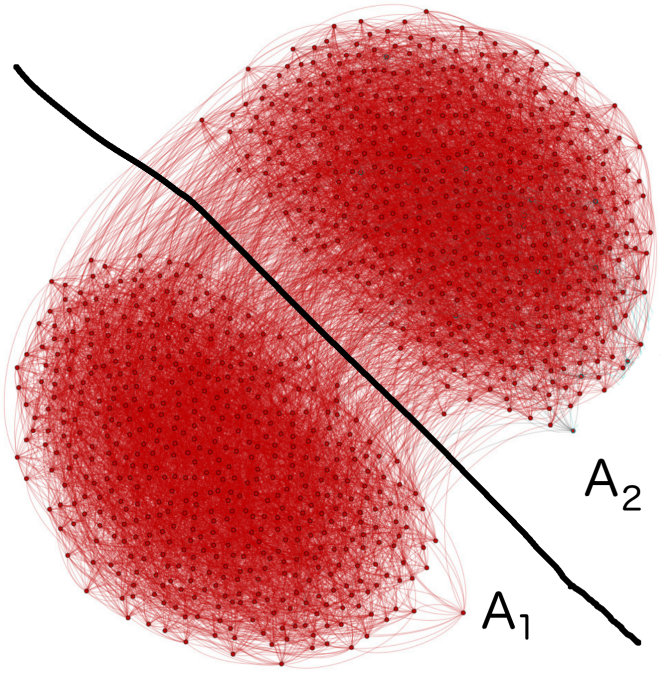
over all partitions $(A_1, \ldots, A_R)$ of the vertex set V



What's wrong with the cut objective?







$$\mathrm{Cond}(A_1) = \frac{\mathrm{Cut}(A_1, A_1^c)}{\mathrm{Vol}(A_1)} = 1$$

$$\mathrm{Cond}(A_2) = \frac{\mathrm{Cut}(A_2, A_2^c)}{\mathrm{Vol}(A_2)} \approx 0$$

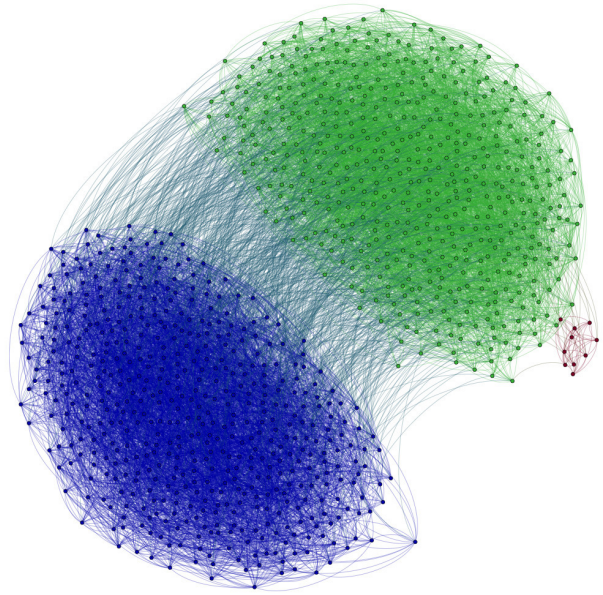
average. cond $= \frac{1}{2}$

$$\mathrm{Cond}(A_1) = \frac{\mathrm{Cut}(A_1, A_1^c)}{\mathrm{Vol}(A_1)} \ll \frac{1}{2}$$

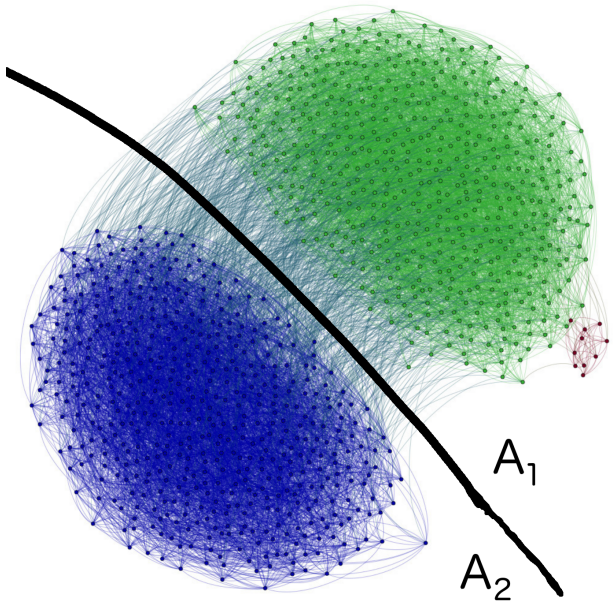
$$\mathrm{Cond}(A_2) = \frac{\mathrm{Cut}(A_2, A_2^c)}{\mathrm{Vol}(A_2)} \ll \frac{1}{2}$$

average. cond $\ll \frac{1}{2}$

What's wrong with the Normalized Cut objective?



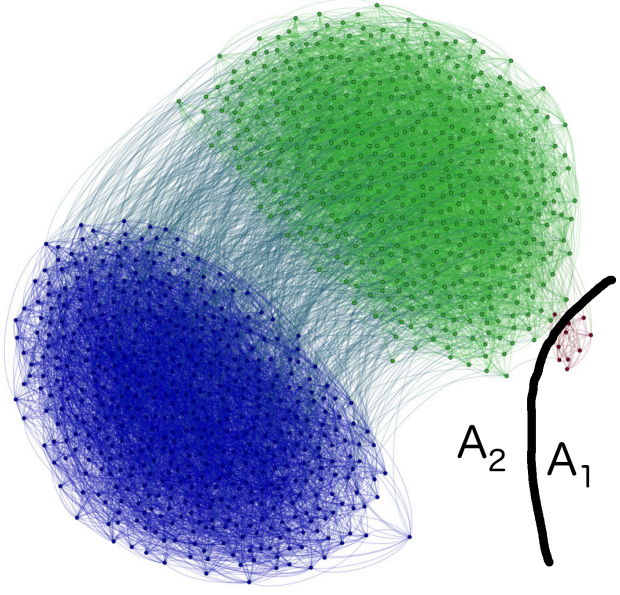
- $A =$ blue cluster. $|A| = n$ $\mathrm{Cond}(A) = \mu$
- $B =$ green cluster. $|B| = n$ $\mathrm{Cond}(B) \approx \mu$
- $C =$ red cluster. $|C| = n_0 \ll n$ $\mathrm{Cond}(C) = \mu_0$



$\mathrm{Cond}(A_1) = \mu$

$\mathrm{Cond}(A_2) \approx \mu$

average. cond $\approx \mu$



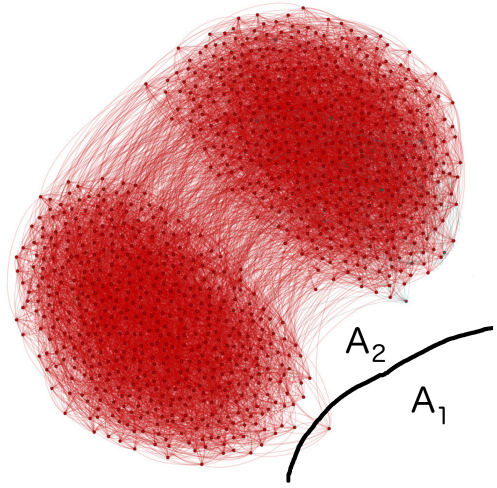
$\mathrm{Cond}(A_1) = \mu_0$

$\mathrm{Cond}(A_2) \approx 0$

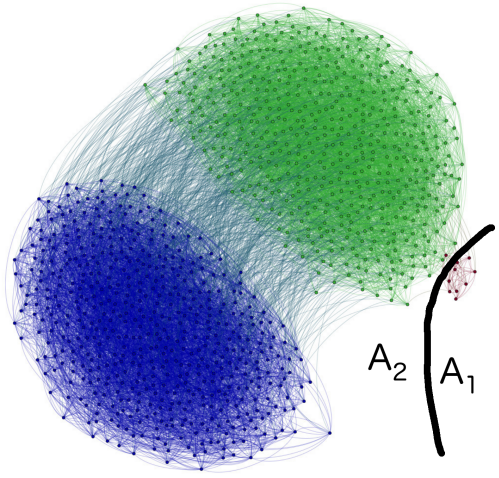
average. cond $\approx \mu_0/2$

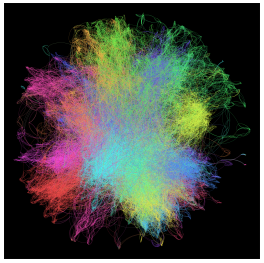
Conclusion

The **Cut objective** is bad because it will cut out outlier vertices



The **Normalized Cut objective** is bad because it will cut out outlier mini-cluster

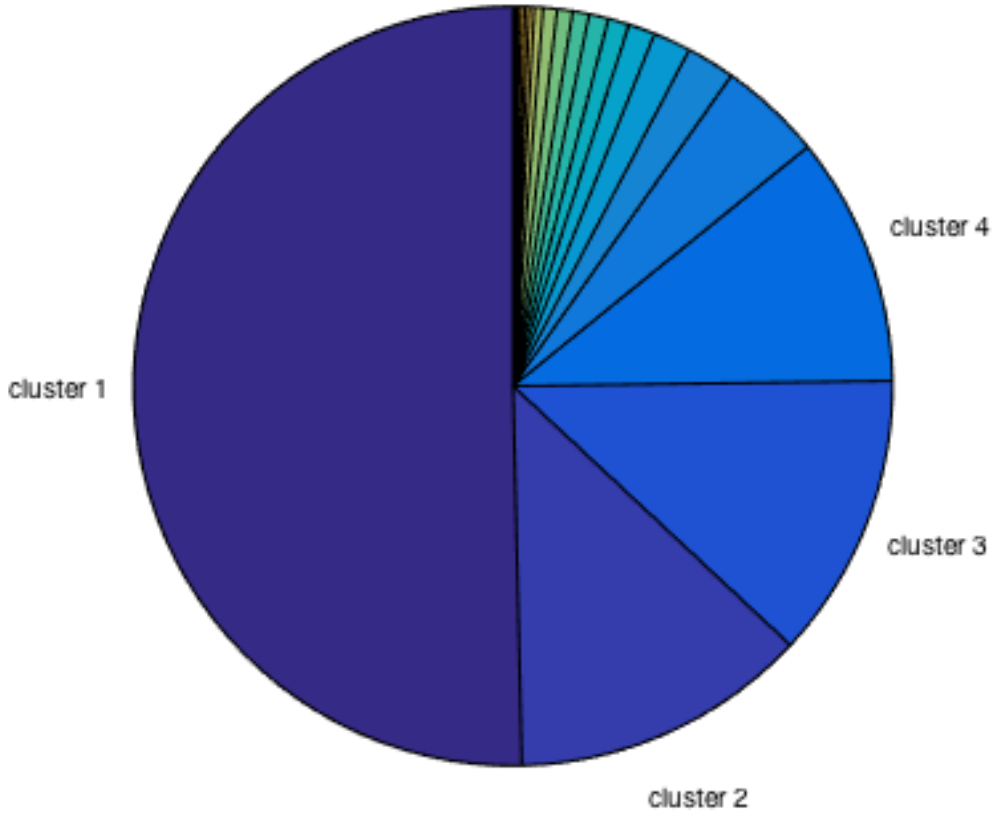




nb of vertices = 20,000
nb of classes= 20

Size of the clusters obtained by NCUT (spectral clustering)

	nb of vertices
cluster 1	10,041
cluster 2	2,527
cluster 3	2,425
cluster 4	2,119
⋮	⋮
⋮	⋮
cluster 16	35
cluster 17	34
cluster 18	27
cluster 19	23
cluster 20	17



proposed model:

The Product Cut

Personalized pagerank vectors

Personalized pagerank vector associated with set A

The vector $\mathbf{pr}_A$ is the solution of

$$\left(\mathrm{Id} + \frac{\alpha}{1-\alpha} L\right) u = \frac{\mathbf{1}_A}{|A|}$$

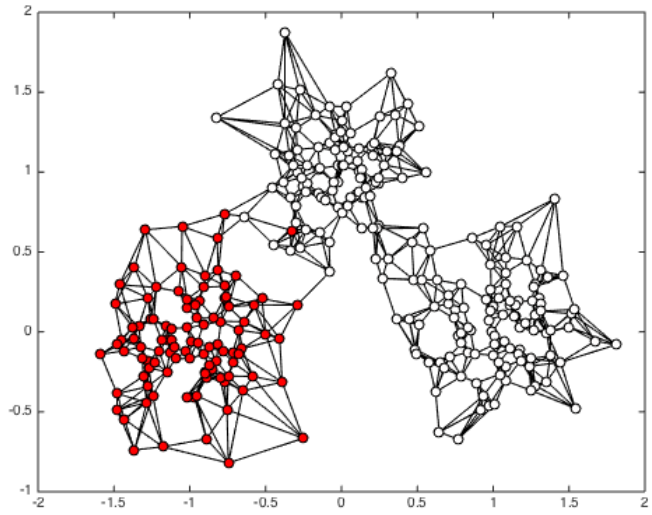
where $L = \mathrm{Id} - WD^{-1}$ is the graph Laplacian matrix.

- $\frac{\mathbf{1}_A}{|A|}$: indicator function A (normalized to have mass 1).
- $\mathbf{pr}_A$: obtained by diffusing $\frac{\mathbf{1}_A}{|A|}$ (also has mass 1).

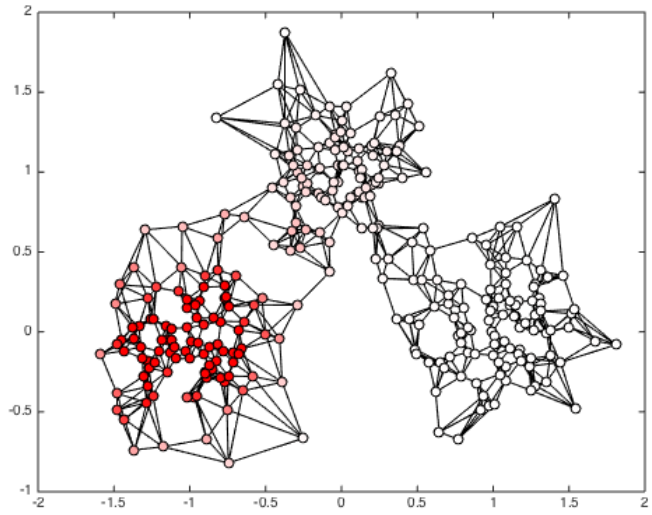
Heat eqn interpretation: This is like doing one implicit step of heat equation with initial data $u^{(0)} = \frac{\mathbf{1}_A}{|A|}$

$$u^{(1)} = u^{(0)} + dt\Delta u^{(1)}$$

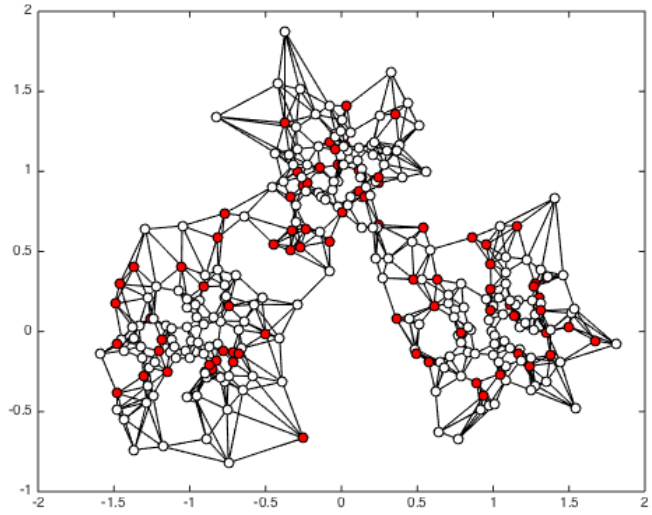
$$(\mathrm{Id} - dt\Delta)u^{(1)} = u^{(0)}$$



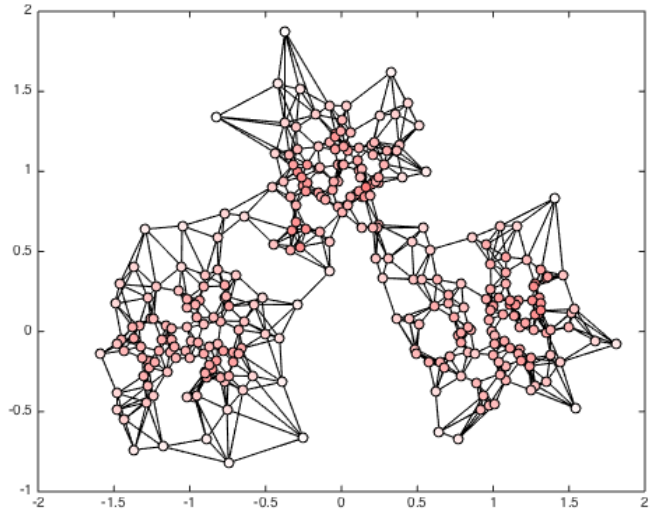
(a) $\mathbf{1}_A/|A|$



(b) $\mathbf{pr}_A$



(c) $\mathbf{1}_A/|A|$



(d) $\mathbf{pr}_A$

Two objectives

Objective 1: Arithmetic average of the pagerank vectors

Maximize $\frac{1}{n}\sum_{r}\sum_{v_i \in A_r} \mathbf{pr}_{A_r}(v_i)$ over all partitions $(A_1, \ldots, A_R)$ of V

Objective 2: Geometric average of the pagerank vectors

Maximize $\left(\prod_{r}\prod_{v_i \in A_r} \mathbf{pr}_{A_r}(v_i)\right)^{1/n}$ over all partitions $(A_1, \ldots, A_R)$ of V

Both objectives aim at finding a partition $(A_1, \ldots, A_R)$ such that

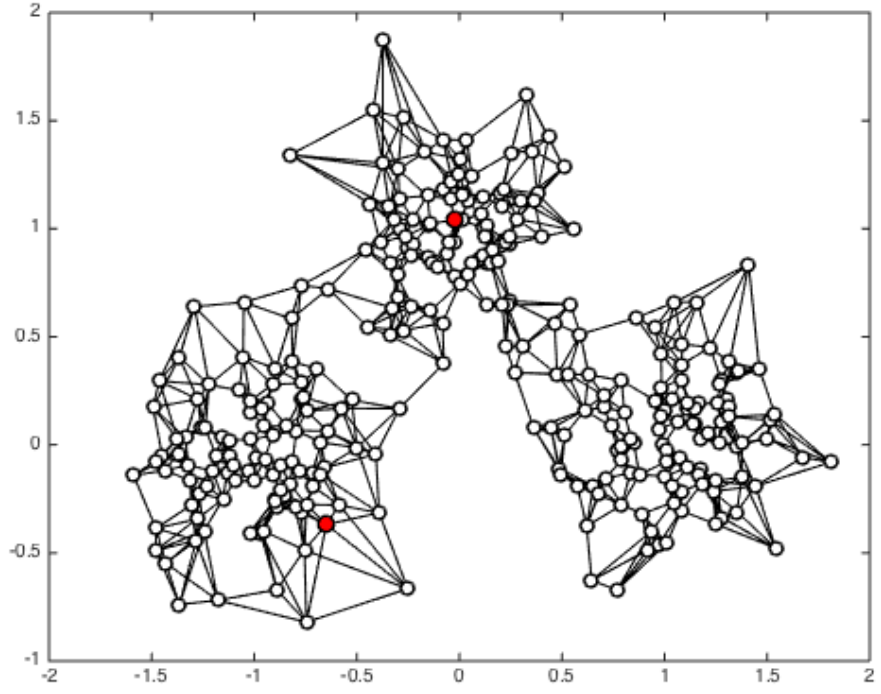
$\mathbf{pr}_{A_r}$ has high values on A_r

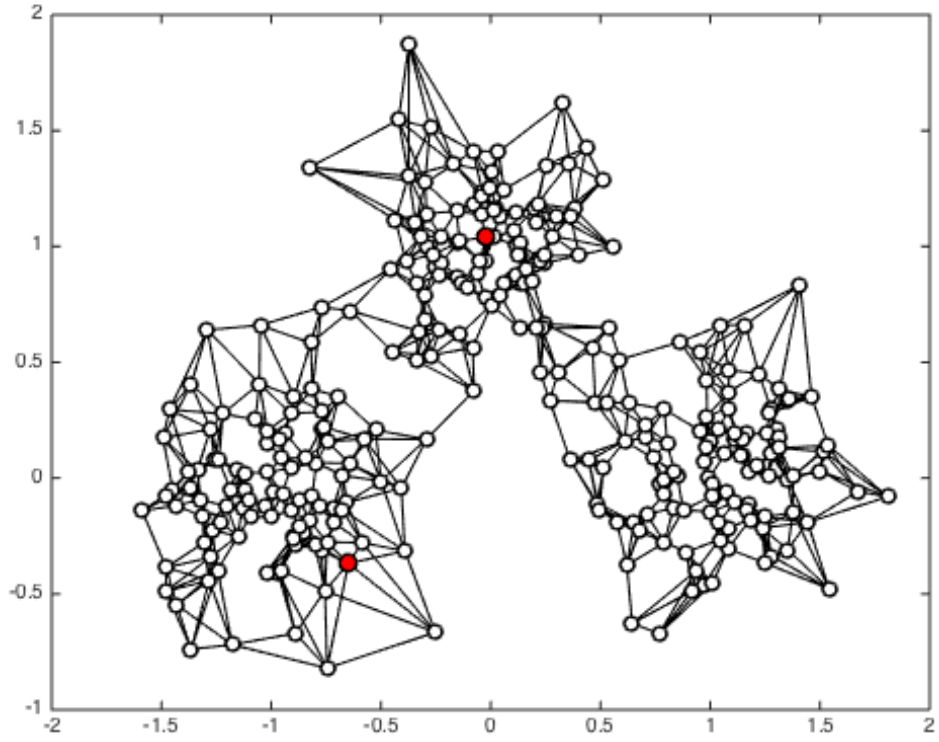
The smooth similarity matrix

Let $\boxed{\Omega = \left(\mathrm{Id} + \frac{\alpha}{1-\alpha} L\right)^{-1}}$

$$\boxed{\omega_{ij} = \mathbf{pr}_{\{v_j\}}(v_i)}$$

ω_{ij} is a nonlocal measure of similarity between vertex v_i and vertex v_j





Original sim. mat. $W = \{w_{ij}\}_{i,j=1}^{n}$

- Sparse
- Symmetric

Smooth sim. mat. $\Omega = \{\omega_{ij}\}_{i,j=1}^{n}$

- Full
- Not symmetric
- column stochastic
- Positive definite

Let $\mathcal{P} = (A_1, \ldots, A_R)$ be a partition of V.

Normalized Cut on the smooth graph

$$\textbf{Ncut}(\mathcal{P}) = \frac{1}{R} \sum_r \frac{\mathrm{Cut}(A_r, A_r^c)}{\mathrm{Vol}(A_r)}$$

cut term: $$\mathrm{Cut}(A_r, A_r^c) = \sum_{i \in A_r} \sum_{j \in A_r^c} \omega_{ij}$$

balance term: $$\mathrm{Vol}(A_r) = \sum_{i \in A_r} d_i$$

$$\text{where } d_i = \sum_{j \in V} \omega_{ji}$$

Objective 1: Arithmetic average of the pagerank vectors

Maximize $\frac{1}{n}\sum_{r}\sum_{v_i \in A_r} \mathbf{pr}_{A_r}(v_i)$ over all partitions $(A_1, \ldots, A_R)$ of V

is equivalent to

Normalized Cut on smooth graph

Minimize $\mathbf{Ncut}(\mathcal{P})$ over all partitions $\mathcal{P} = (A_1, \ldots, A_R)$ of V

Objective 2: Geometric average of the pagerank vectors

Maximize $\left(\prod_{r} \prod_{v_i \in A_r} \mathbf{pr}_{A_r}(v_i) \right)^{1/n}$ over all partitions $(A_1, \ldots, A_R)$ of V

is equivalent to

Product Cut

Minimize $\mathbf{Pcut}(\mathcal{P})$ over all partitions $\mathcal{P} = (A_1, \ldots, A_R)$ of V

Let $\mathcal{P} = (A_1, \ldots, A_R)$ be a partition of V.

Product Cut

$$\textbf{Pcut}(\mathcal{P}) = \frac{\prod_{r=1}^{R} \mathcal{Z}(A_r, A_r^c)^{1/n}}{e^{H(\mathcal{P})}}$$

cut term:

$$\boxed{\mathcal{Z}(A, A^c) := \prod_{v_i \in A} \left(1 + \frac{\sum_{j \in A^c} \omega_{ij}}{\sum_{j \in A} \omega_{ij}}\right)}$$

$\mathcal{Z}(A, A^c) = 1$ if A and A^c are disconnected
$\mathcal{Z}(A, A^c) \approx 2^{n_r}$ if A and A^c are completely mixed

balance term:

$$\boxed{H(\mathcal{P}) := -\sum_{r=1}^{R} \theta_r \log \theta_r} \qquad \text{where } \theta_r = \frac{|A_r|}{|V|}$$

$H(\mathcal{P}) = 0$ if $\mathcal{P} = (V, \emptyset, \ldots, \emptyset)$
$H(\mathcal{P})$ is maximal if $|A_1| = |A_2| = \ldots = |A_R|$

Normalized Cut

$$\mathbf{Ncut}(\mathcal{P}) = \frac{1}{R}\sum_{r}\frac{\mathrm{Cut}(A_r, A_r^c)}{\mathrm{Vol}(A_r)}$$

$$0 \leq \mathbf{Ncut}(\mathcal{P}) \leq 1$$

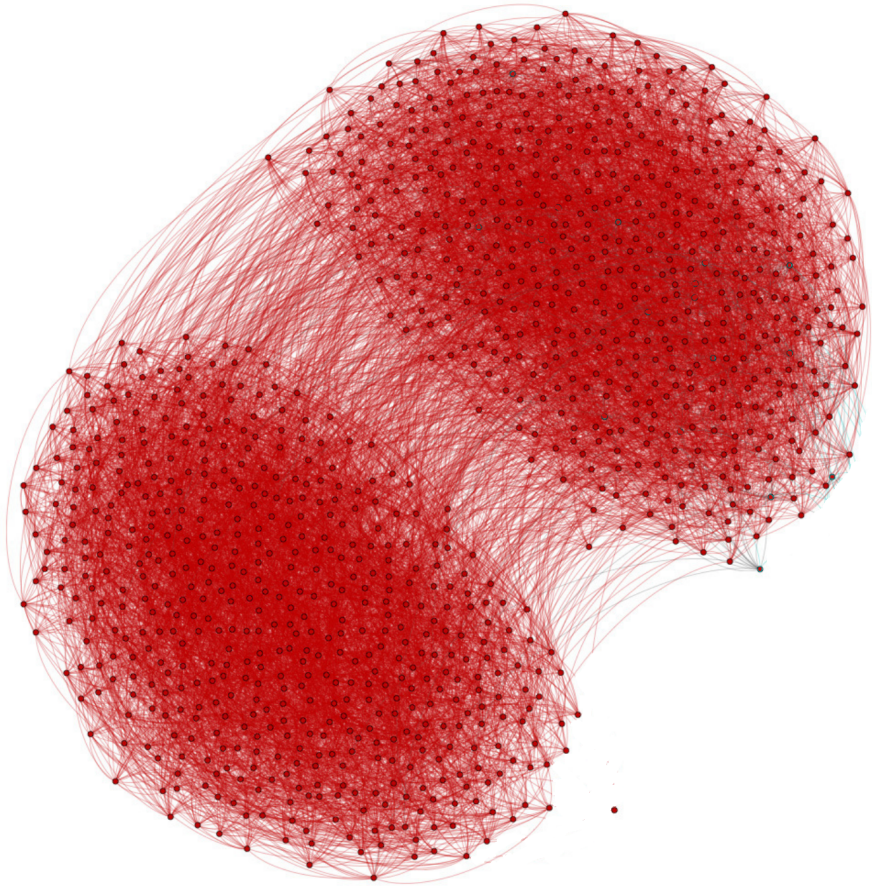
Product Cut

$$\mathbf{Pcut}(\mathcal{P}) = \frac{\prod_{r=1}^{R}\mathcal{Z}(A_r, A_r^c)^{1/n}}{e^{H(\mathcal{P})}}$$

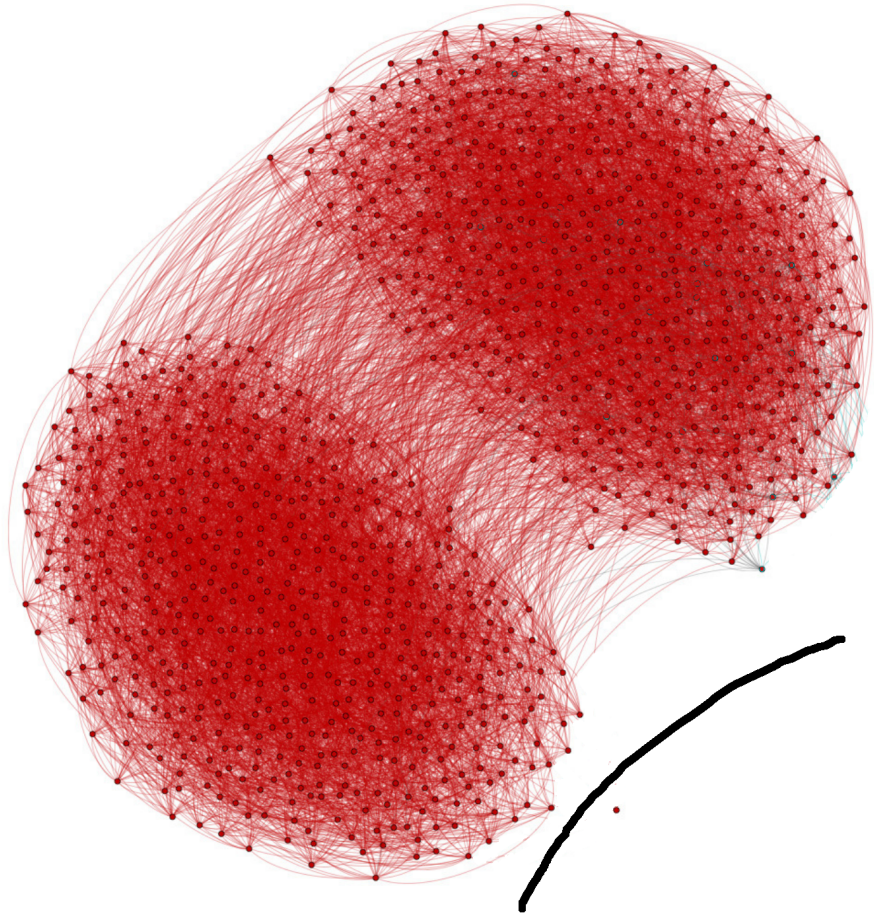
$$e^{-H(\mathcal{P})} \leq \mathbf{Pcut}(\mathcal{P}) \leq 1$$

$$\lim_{k\to\infty} H(\mathcal{P}^{(k)}) = 0 \quad \Rightarrow \quad \lim_{k\to\infty} \mathbf{Pcut}(\mathcal{P}^{(k)}) = 1.$$

Example 1: Disconnected graph

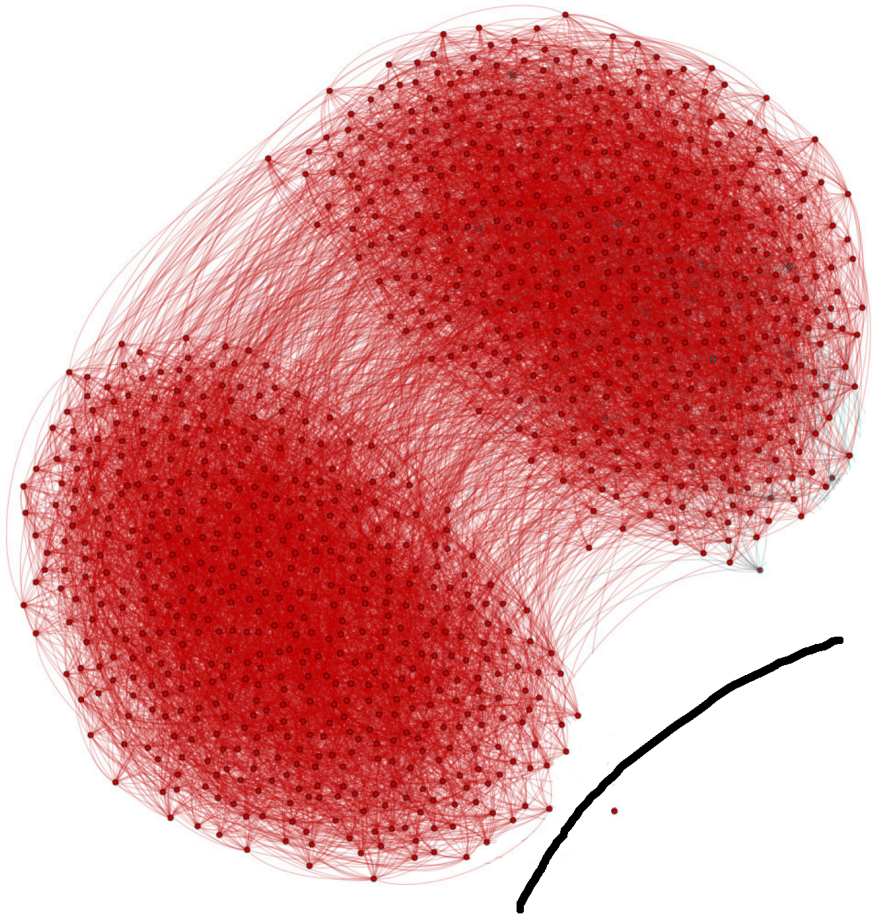


Example 1: Disconnected graph



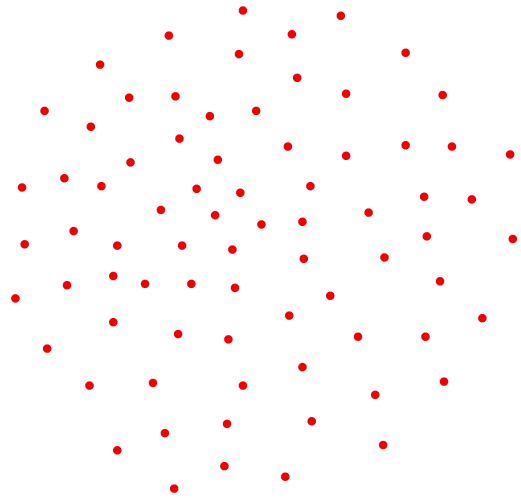
$$\textbf{Ncut}(\mathcal{P}) = \frac{1}{R}\sum_{r}\frac{\mathrm{Cut}(A_r, A_r^c)}{\mathrm{Vol}(A_r)} = 0 \qquad \text{(best possible value!)}$$

Example 1: Disconnected graph



$$\textbf{Pcut}(\mathcal{P}) = \frac{\prod_{r=1}^{R} \mathcal{Z}(A_r, A_r^c)^{1/n}}{e^{H(\mathcal{P})}} \approx \frac{1}{e^0} = 1 \qquad \text{(worst possible value!)}$$

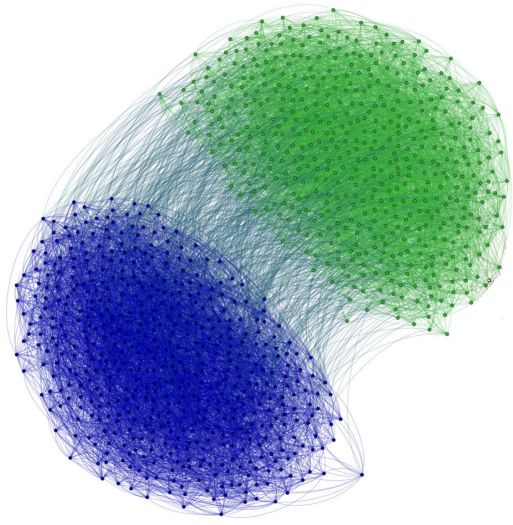
Example 2: Fully disconnected graph



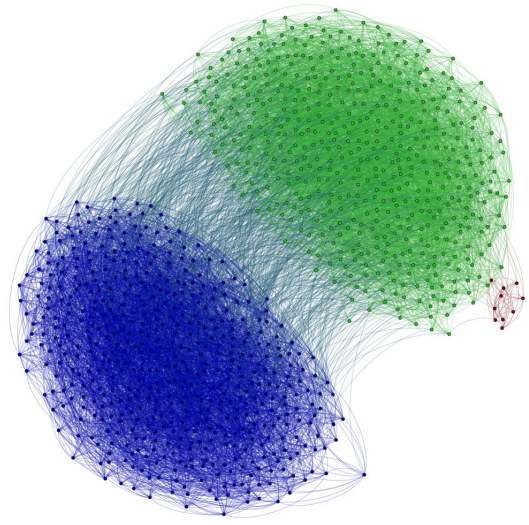
- **Ncut**$(\mathcal{P}) = \dfrac{1}{R}\displaystyle\sum_{r} \dfrac{\mathrm{Cut}(A_r, A_r^c)}{\mathrm{Vol}(A_r)} = 0$ for all partitions $\mathcal{P}$

- **Pcut**$(\mathcal{P}) = \dfrac{\prod_{r=1}^{R} \mathcal{Z}(A_r, A_r^c)^{1/n}}{e^{H(\mathcal{P})}} = \dfrac{1}{e^{H(\mathcal{P})}}$ for all partitions $\mathcal{P}$

Example 3: Stability with respect to perturbation



(e) Unperturbed Graph



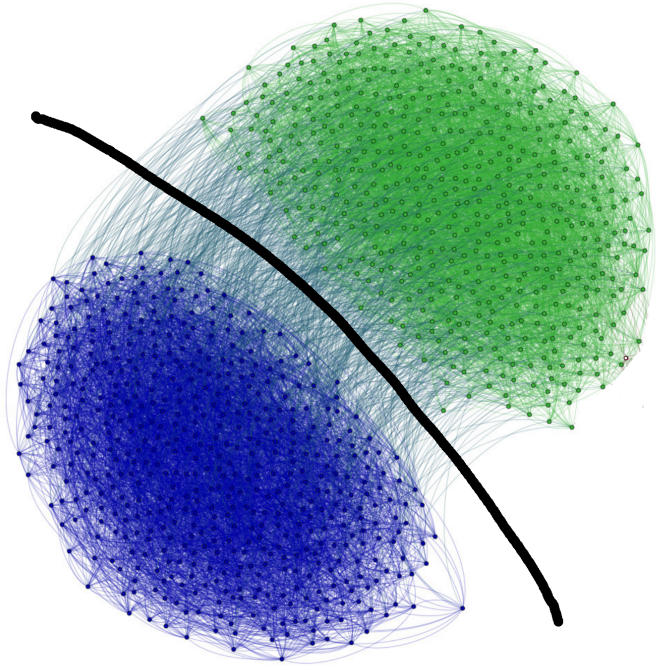
(f) Perturbed Graph

- $A =$ blue cluster. $\quad |A| = n \quad \mathrm{Cond}(A) = \mu$
- $B =$ green cluster. $\quad |B| = n \quad \mathrm{Cond}(B) \approx \mu$
- $C =$ red cluster. $\quad |C| = n_0 \ll n \quad \mathrm{Cond}(C) = \mu_0$

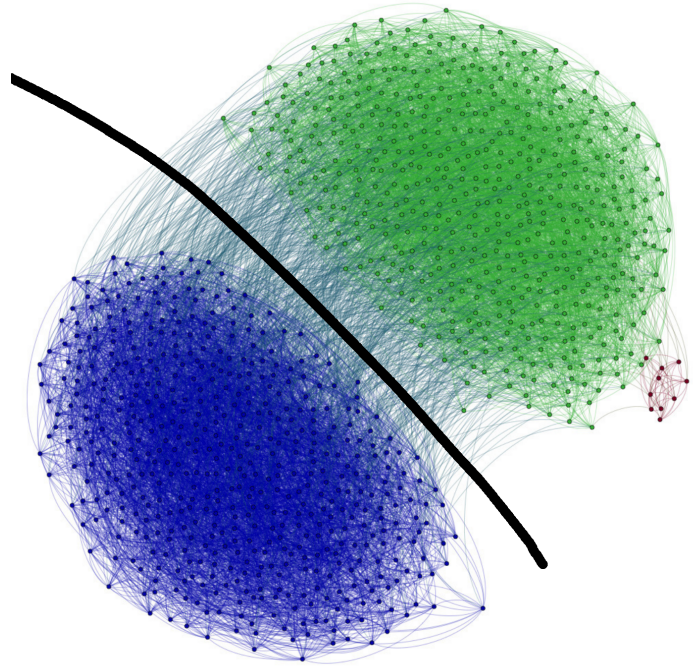
μ, μ_0 and n_0 are fixed
$\mu_0 < 2\mu$
Consider the limit where $n \to \infty$

Pcut is stable with respect to small perturbations

For n large enough we have:



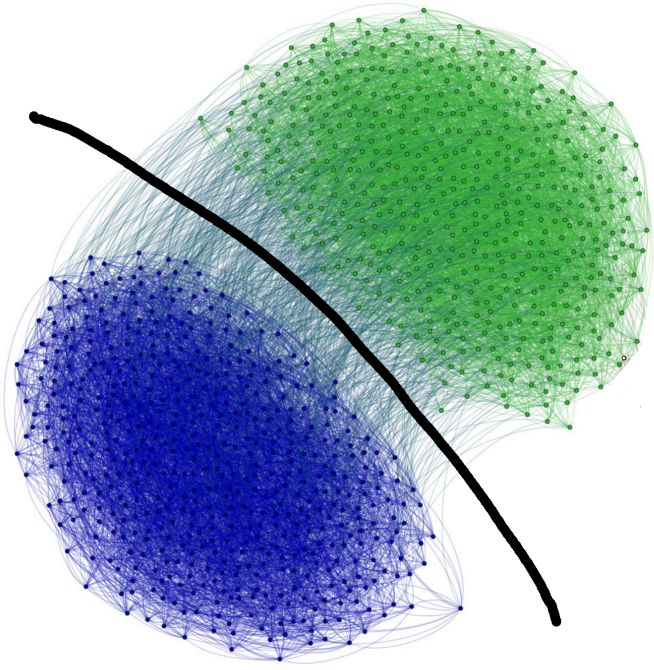
(g) Pcut of unperturbed Graph



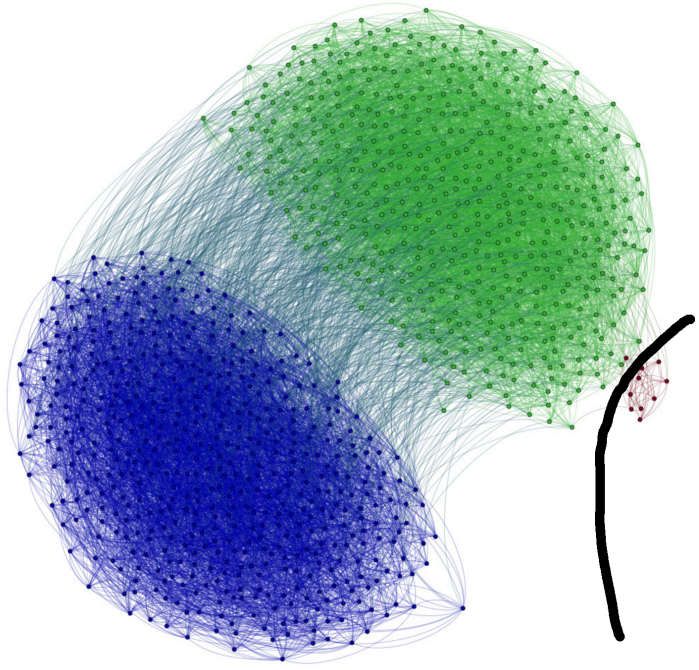
(h) Pcut of perturbed Graph

Ncut is not stable with respect to small perturbations

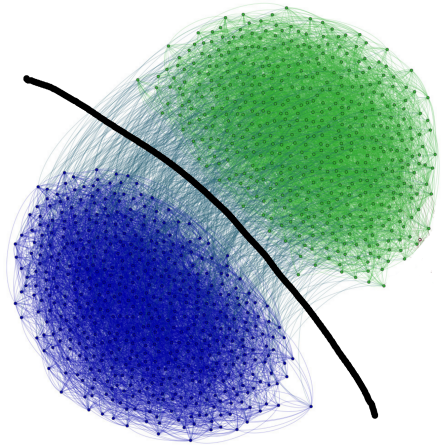
For n large enough we have:



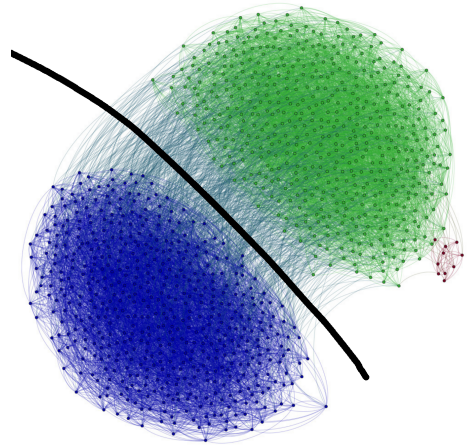
(i) Ncut of unperturbed Graph



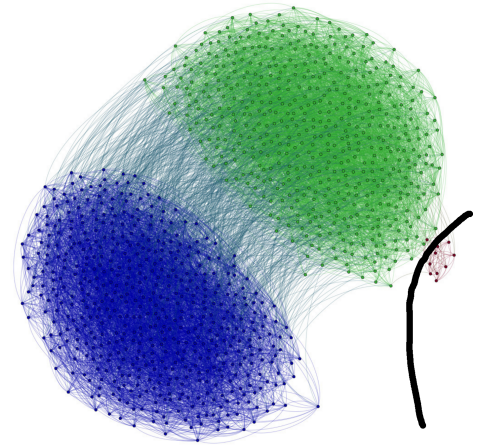
(j) Ncut of perturbed Graph



(k) $\mathcal{P}_n^{\text{good}}$



(l) $\mathcal{P}_n^{0,\text{good}}$



(m) $\mathcal{P}_n^{0,\text{bad}}$

The inequality $e^{-H(\mathcal{P})} \leq \mathbf{Pcut}(\mathcal{P}) \leq 1$ implies:

$$\lim_{n\to\infty} \mathbf{Pcut}_{\mathcal{G}_n^0}(\mathcal{P}_n^{0,\text{bad}}) = 1.$$

The unperturbed graph $\mathcal{G}_n$ grows in a self-similar fashion as $n \to \infty$:

$$\mathbf{Pcut}_{\mathcal{G}_n}(\mathcal{P}_n^{\text{good}}) \approx \gamma < 1 \qquad \text{for all } n$$

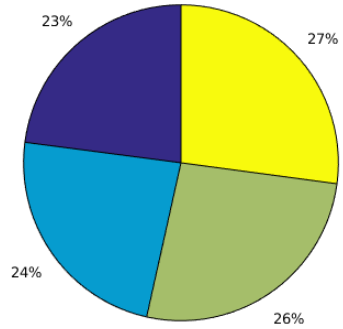
The perturbation is infinitesimally small so:

$$\mathbf{Pcut}_{\mathcal{G}_n^0}(\mathcal{P}_n^{0,\text{good}}) \approx \mathbf{Pcut}_{\mathcal{G}_n}(\mathcal{P}_n^{\text{good}}) \approx \gamma < 1$$

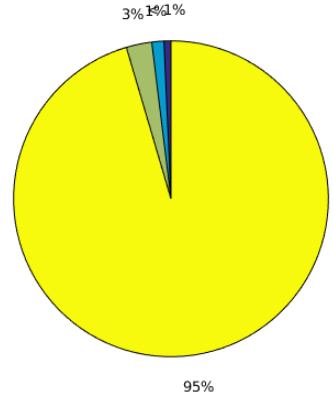
So for n large enough

$$\mathbf{Pcut}_{\mathcal{G}_n^0}(\mathcal{P}_n^{0,\text{good}}) < \mathbf{Pcut}_{\mathcal{G}_n^0}(\mathcal{P}_n^{0,\text{bad}})$$

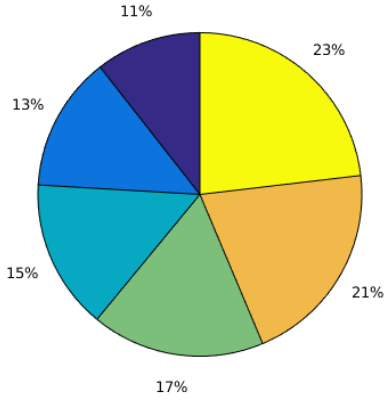
Example 4: Real world datasets



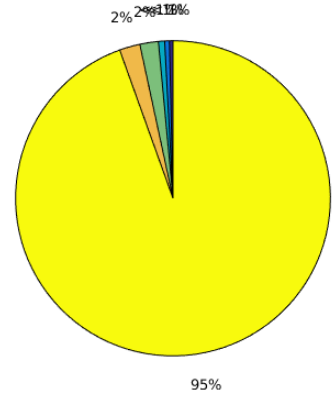
(n) **Pcut** of WEBKB4



(o) **Ncut** of WEBKB4



(p) **Pcut** of CITESEER



(q) **Ncut** of CITESEER

Exact Continuous relaxation

and

Algorithm

$$\left.\begin{array}{l}\text{Minimize} \quad \mathbf{Pcut}(\mathcal{P}) \\ \text{over all partitions } \mathcal{P} = (A_1, \ldots, A_R) \text{ of } V\end{array}\right\} \quad \text{(P1)}$$

is equivalent to

$$\left.\begin{array}{l}\text{Maximize} \quad \dfrac{1}{n}\displaystyle\sum_{r=1}^{R}\left\langle \mathbf{1}_{A_r} \ , \ \log\dfrac{\Omega\,\mathbf{1}_{A_r}}{|A_r|}\right\rangle \\ \text{over all partitions } \mathcal{P} = (A_1, \ldots, A_R) \text{ of } V\end{array}\right\} \quad \text{(P2)}$$

continuous relaxation

$$\left.\begin{array}{l}\text{Maximize} \quad \dfrac{1}{n}\displaystyle\sum_{r=1}^{R}\left\langle f_r \ , \ \log\dfrac{\Omega\, f_r}{\langle f_r, \mathbf{1}_V\rangle}\right\rangle \\ \\ \text{over all } f_1, \ldots, f_R : V \to [0, +\infty) \\ \\ \text{such that } \displaystyle\sum_{r=1}^{R} f_r(v_i) = 1\end{array}\right\} \quad \text{(P3)}$$

$$\left.\begin{array}{l} \text{Maximize} \quad \dfrac{1}{n}\displaystyle\sum_{r=1}^{R}\left\langle \mathbf{1}_{A_r} \ , \ \log\frac{\Omega\,\mathbf{1}_{A_r}}{|A_r|}\right\rangle \\ \text{over all partitions } \mathcal{P}=(A_1,\ldots,A_R) \text{ of } V \end{array}\right\} \qquad \text{(P2)}$$

continuous relaxation

$$\left.\begin{array}{l} \text{Maximize} \quad \dfrac{1}{n}\displaystyle\sum_{r=1}^{R}\left\langle f_r \ , \ \log\frac{\Omega\, f_r}{\langle f_r, \mathbf{1}_V\rangle}\right\rangle \\ \text{over all } f_1,\ldots,f_R : V \to [0,+\infty) \\ \text{such that } \displaystyle\sum_{r=1}^{R} f_r(v_i) = 1 \end{array}\right\} \qquad \text{(P3)}$$

Main Theorem

The functional $e(f)=\left\langle f \ , \ \log\dfrac{\Omega\, f}{\langle f, \mathbf{1}_V\rangle}\right\rangle$ is convex!

As a consequence (P2) and (P3) are equivalent.

So computing the Product Cut of a graph is equivalent to a convex maximization problem over the simplex:

The Product Cut can be written

$$\begin{array}{ll} \text{Maximize} & \mathcal{E}(F) \\ \text{subject to} & F \in C \\ & \psi_i(F) = 0 \text{ for } i = 1, \ldots, n \end{array}$$

where

- $F = (f_1, \ldots, f_R)$
- $C = \mathbb{R}_+^n \times \ldots \times \mathbb{R}_+^n$
- $\psi_i(F) = \left(\sum_{r=1}^{R} f_r(v_i) \right) - 1$

Algorithm 1 (Sequential Linear Programming)

The next iterate $F^{(k+1)}$ is obtained by solving the Linear Program

$$\begin{aligned} &\text{Maximize} && \mathcal{L}_k(F) \\ &\text{subject to} && F \in C \\ & && \psi_i(F) = 0 \text{ for } i = 1, \ldots, n \end{aligned}$$

where

$$\mathcal{L}_k(F) = \mathcal{E}(F^k) + \langle \nabla \mathcal{E}(F^{(k)}), F - F^k \rangle$$

is the linearization of the energy $\mathcal{E}(F)$ around the current iterate.

Lemma (the successive iterates are indicator functions of partitions)

$F^{(k)} = (\mathbf{1}_{A_1}, \ldots, \mathbf{1}_{A_R})$ *for some partitions* $(A_1, \ldots, A_R)$ *of* V.

Lemma (Monotonicity)

$\mathcal{E}(F^{(k+1)}) \geq \mathcal{E}(F^{(k)})$

Proof: Since $\mathcal{E}(F)$ is convex we have:

$$\mathcal{E}(F^{k+1}) \geq \mathcal{L}_k(F^{k+1}) \geq \mathcal{L}_k(F^k) = \mathcal{E}(F^k).$$

Algorithm 2 (Randomized Sequential Linear Programming)

The next iterate $F^{(k+1)}$ is obtained by solving the Linear Program

$$\begin{aligned} \text{Maximize} \quad & \mathcal{L}_k(F) \\ \text{subject to} \quad & F \in C \\ & \psi_i(F) = 0 \text{ for } i \in \mathcal{I}_k \end{aligned}$$

where

$$\mathcal{I}_k \text{ is a random subset of } \{1, 2, \ldots, n\} \text{ of size } s_k$$

The vertices not in $\mathcal{I}_k$ are allowed to belong to multiple classes or no class at all.

We need a schedule to determine the rate at which s_k increases:

- Start with $s_0 \approx 1\%$ of n.
- Then s_k increases linearly until $s_k \approx 30\%$ of n
- Then set $s_k = n$ until the algorithm converges.

Algorithm 2 (Randomized Sequential Linear Programming)

The next iterate $F^{(k+1)}$ is obtained by solving the Linear Program

$$\begin{aligned} \text{Maximize} \quad & \mathcal{L}_k(F) \\ \text{subject to} \quad & F \in C \\ & \psi_i(F) = 0 \text{ for } i \in \mathcal{I}_k \end{aligned}$$

$$\mathcal{L}_k(F) = \mathcal{E}(F^k) + \langle \nabla \mathcal{E}(F^{(k)}), F - F^k \rangle$$

Algorithm 2 (Randomized Sequential Linear Programming)

The next iterate $F^{(k+1)}$ is obtained by solving the Linear Program

$$\begin{aligned} \text{Maximize} \quad & \langle \nabla \mathcal{E}(F^{(k)}), F \rangle \\ \text{subject to} \quad & F \in C \\ & \psi_i(F) = 0 \text{ for } i \in \mathcal{I}_k \end{aligned}$$

This Linear program has an explicit solution obtained by thresholding $\nabla \mathcal{E}(F^{(k)})$

Algorithm 2 (Randomized Sequential Linear Programming)

- Compute the n-by-R matrix $\nabla\mathcal{E}(F^{(k)}) = [h_1, \ldots, h_R]$

$$h_r = \log\left(\mathbf{pr}_{A_r}\right) + v_r - \mathbf{1}_V$$

- Choose at random s_k vertices and let $\mathcal{I}_k \subset V$ be these vertices.

- Threshold the matrix $\nabla\mathcal{E}(F^{(k)})$

$$\text{If } i \in \mathcal{I}_k, \text{ then} \qquad f_{i,r}^{(k+1)} = \begin{cases} 1 & \text{if } r = \arg\max_s h_{is} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{If } i \notin \mathcal{I}_k, \text{ then} \qquad f_{i,r}^{(k+1)} = \begin{cases} 1 & \text{if } h_{i,r} > 0 \\ 0 & \text{otherwise} \end{cases}$$

It is an MBO like algorithm which alternate between
1) Diffusing characteristic functions and 2) Thresholding.

Dominant cost of the algorithm

At each step of the algorithm we need to compute the R personalized pagerank vectors $\mathbf{pr}_{A_r}$.

Personalized pagerank vector associated with set A

The vector $\mathbf{pr}_A$ is the solution of

$$\left(\mathrm{Id} + \frac{\alpha}{1-\alpha} L\right) u = \frac{\mathbf{1}_A}{|A|}$$

We use Algebraic Multigrid technique in order to solve this as fast and as approximatively as possible!

Oren Livne and Achi Brandt.
Lean algebraic multigrid (lamg): Fast graph laplacian linear solver.

Algorithmic Comparison via Cluster Purity

	20NE	RCV1	WEBK	CITE	MNIS	PEND	USPS	OPTI
size	20K	9.6K	4.2K	3.3K	70K	11K	9.3K	5.6K
R	20	4	4	6	10	10	10	10
NCUT	27	38	40	23	77	80	72	91
LSD	34	38	46	53	76	86	70	91
MTV	36	43	45	43	96	**87**	85	95
GRACLUS	42	42	49	54	**97**	85	87	94
NMFR	**61**	43	**58**	**63**	**97**	**87**	86	**98**
PCut	**61**	**53**	**58**	**63**	**97**	**87**	**89**	**98**

Computational Time

MNIST			**20NEWS**		
NMFR	**PCut** $(.9,\lambda_1)$	**PCut** $(.9,\lambda_2)$	**NMFR**	**PCut** $(.9,\lambda_1)$	**PCut** $(.9,\lambda_2)$
4.6mn	11s	10s	3.7mn	1.3mn	16s
(92%)	(92%)	(91%)	(58%)	(58%)	(57%)

Conclusion

- We have introduce a novel graph partitioning objective:

$$\mathbf{Pcut}(\mathcal{P}) = \frac{\prod_{r=1}^{R} \mathcal{Z}(A_r, A_r^c)^{1/n}}{e^{H(\mathcal{P})}}$$

- This objective has nice mathematical properties:
  1. $e^{-H(\mathcal{P})} \leq \mathbf{Pcut}(\mathcal{P}) \leq 1$
  2. Convexity of the continuous relaxation $e(f)$.
  3. Equivalent to a convex maximization problem over the simplex

- The algorithm achieves state of the art on benchmark datasets.

Thanks