Four graph partitioning algorithms

Fan Chung

University of California, San Diego

History of graph partitioning

NP-hard → approximating partitioning

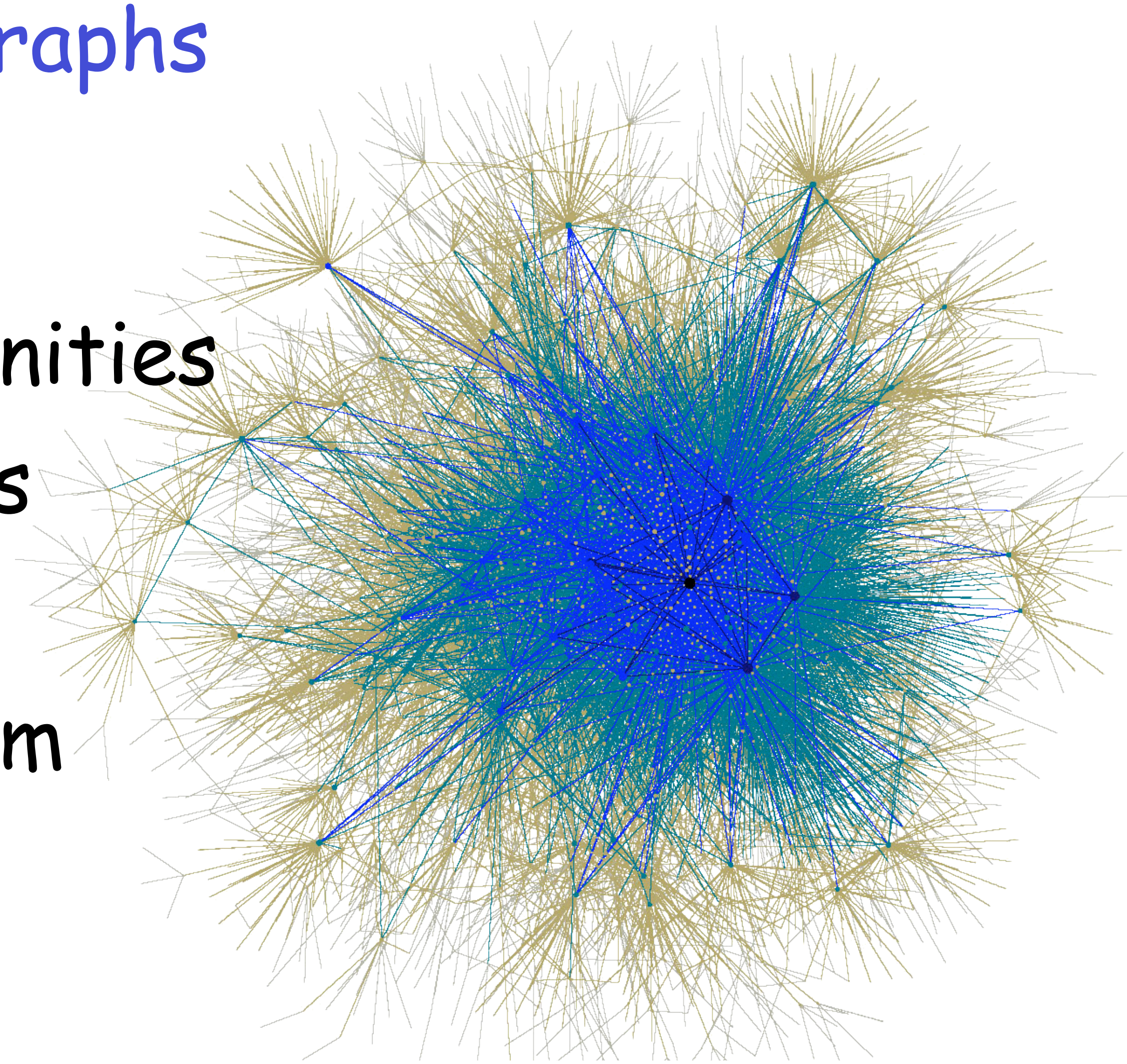
- Eigenvector, Fiedler 73, Folklore,
- Multicommunity flow, Leighton+Rao 88
- Semidefinite program,
  Arora+Rao+Vazirani 04
- Expander flow, Arora+Hazan+Kale 04
- Single commodity flows,
  Khandekar+Rao+Vazirani 06

Usual applications of graph partition algorithms:

- Divide-and-conquer algorithms
- Declustering algorithms
- Circuit layout & designs
- Parallel computing
- Bioinformatics
- …

Applications of partitioning algorithms for massive graphs

- Web search
- identify communities
- locate hot spots
- trace targets
- combat link spam
- epidemics
- …



Ranking Web pages

Partitioning algorithm for massive graphs

Google "network science" Search [Advanced Search](#) [Preferences](#)

Web Results **1** - **10** of about **341,000** for "**network science**". (**0.33** seconds)

Network Science
To help address this problem, the Army asked the National Research Council to find out whether identifying and funding **network science** research could help ...
www.nap.edu/catalog/11516.html - 35k - Cached - Similar pages - Note this

Network Science
Network Science THE NATIONAL ACADEMIES PRESS 500 Fifth Street, N.W. Washington, DC 20001 NOTICE: The project that is the subject of this report was approved ...
www.nap.edu/books/0309100267/html/ - 67k - Cached - Similar pages - Note this

NetSci | Resources for Pharmaceutical Research and Development
Welcome to the **Network Science** website. This site is dedicated to the topics of ... NetSci, ISSN 1092-7360, is published by **Network Science** Corporation. ...
www.netsci.org/ - 7k - Cached - Similar pages - Note this

The NetSci Science Center
This section of the **Network Science** homepage contains articles written by scientists from pharmaceutical and biotechnology companies as well as leading ...
www.netsci.org/Science/ - 7k - Cached - Similar pages - Note this
More results from www.netsci.org »

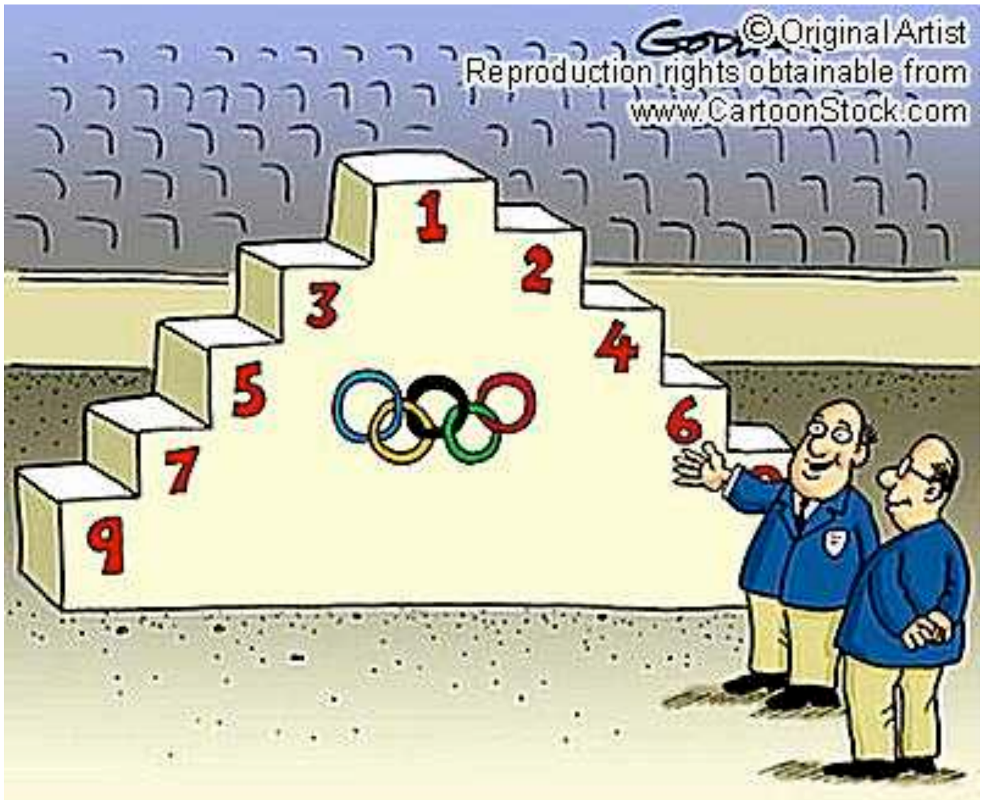
NetSci 07 | International Workshop and Conference on **Network Science**
The International Workshop and Conference on **Network Science** (NetSci07) brings together leading researchers, practitioners, and teachers in **network science** ...
www.nd.edu/~netsci/ - 11k - Cached - Similar pages - Note this

Outline of the talk

- Motivations
- Conductance and Cheeger's inequality
- Four graph paritioning algorithms by using:
  - eigenvectors
  - random walks
  - PageRank
  - heat kernel
- Local graph algorithms
- Future directions

What is PageRank?

What is Rank?



"We're hoping this new podium design will mean
we get to see some British athletes!"

Search Engines:

Ask, and it will be given to you;
seek, and you will find;
knock, and it will be opened to you.

\- Mathew 7:7

眾裡尋他千百度
驀然回首 那人卻在
燈火闌珊處
辛棄疾 青玉案

Home

About Google

Help Central

Google Features

Services & Tools

Our Technology
Why Use Google
Benefits of Google

Find on this site:

Search

Google searches more sites more quickly, delivering the most relevant results.

Introduction

Google runs on a unique combination of advanced hardware and software. The speed you experience can be attributed in part to the efficiency of our search algorithm and partly to the thousands of low cost PC's we've networked together to create a superfast search engine.

The heart of our software is PageRank™, a system for ranking web pages developed by our founders Larry Page and Sergey Brin at Stanford University. And while we have dozens of engineers working to improve every aspect of Google on a daily basis, PageRank continues to play a central role in many of our web search tools.

PageRank Explained

PageRank relies on the uniquely democratic nature of the web by using its vast link structure as an indicator of an individual page's value. In essence, Google interprets a link from page A to page B as a vote, by page A, for page B. But, Google looks at considerably more than the sheer volume of votes, or links a page receives; for example, it also analyzes the page that casts the vote. Votes cast by pages that are themselves "important" weigh more heavily and help to make other pages "important." Using these and other factors, Google provides its

Our Search: Google Technology

Google searches more sites more quickly, delivering the most relevant results.

Introduction

Google runs on a unique combination of advanced hardware and software. The speed you experience can be attributed in part to the efficiency of our search algorithm and partly to the thousands of low cost PC's we've networked together to create a superfast search engine.

The heart of our software is PageRank™, a system for ranking web pages developed by our founders Larry Page and Sergey Brin at Stanford University. And while we have dozens of engineers working to improve every aspect of Google on a daily basis, PageRank continues to play a central role in many of ou web search tools.

PageRank Explained

PageRank relies on the uniquely democratic nature of the web by using its vast link structure as an indicator of an individual page's value. In essence, Google interprets a link from page A to page B as a vote, by page A, for page B. But, Google looks at considerably more than the sheer volume of votes, or links a page receives; for example, it also analyzes the page that casts the vote. Votes cast by pages that are themselves "important" weigh more heavily and help to make other pages "important." Using these and other factors, Google provides its

What is PageRank?

Google's answer:

PageRank Explained

PageRank relies on the uniquely democratic nature of the web by using its vast link structure as an indicator of an individual page's value. In essence, Google interprets a link from page A to page B as a vote, by page A, for page B. But, Google looks at considerably more than the sheer volume of votes, or links a page receives; for example, it also analyzes the page that casts the vote. Votes cast by pages that are themselves "important" weigh more heavily and help to make other pages "important." Using these and other factors, Google provides its views on pages' relative importance.

What is PageRank?

PageRank is a well-defined operator on any given graph, introduced by Sergey Brin and Larry Page of Google in a paper of 1998.

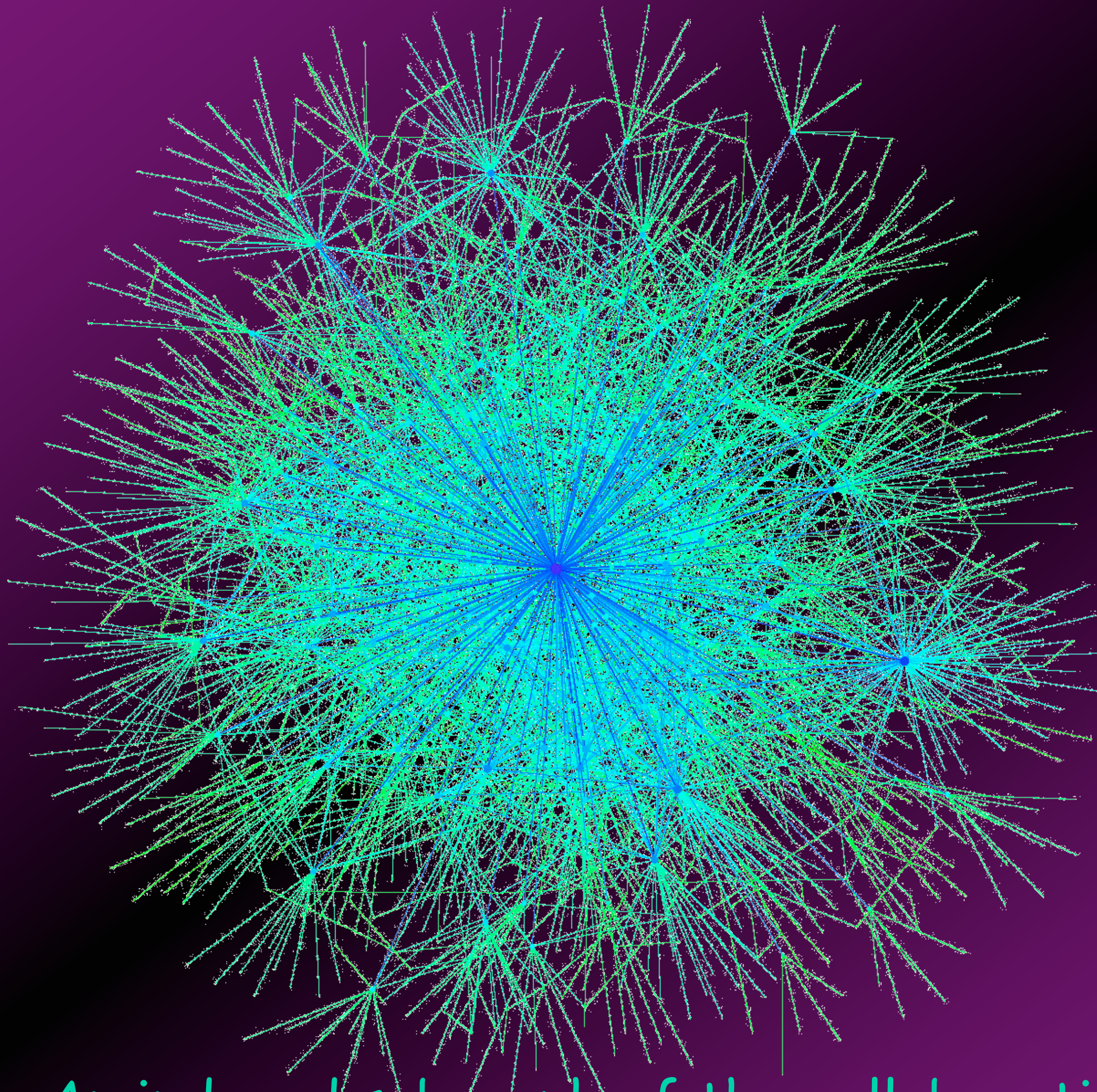
Graph models

Vertices	Edges
cities	flights
people	pairs of friends
telephones	phone calls
web pages	linkings
genes	regulatory effect

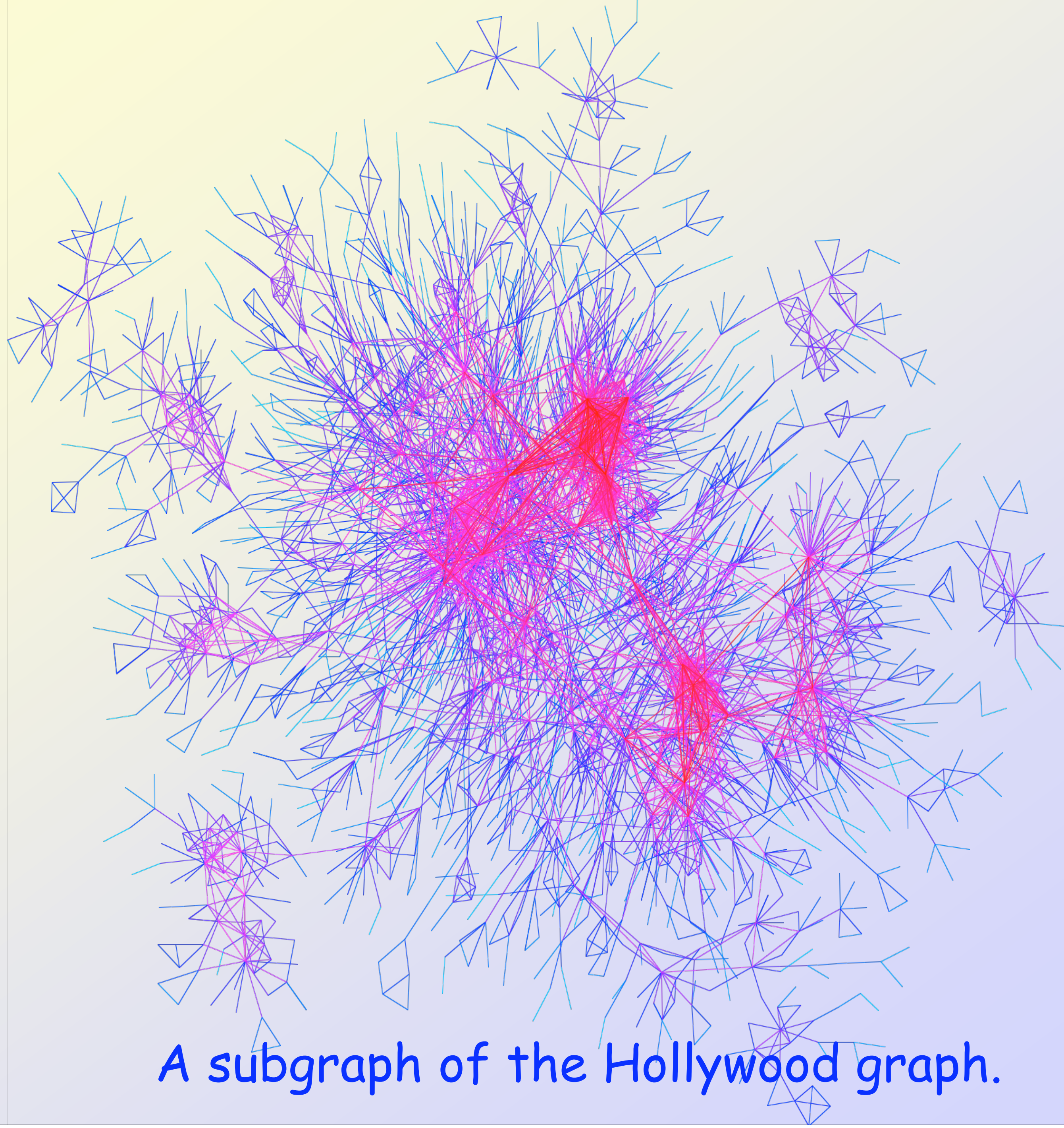
Information

relations

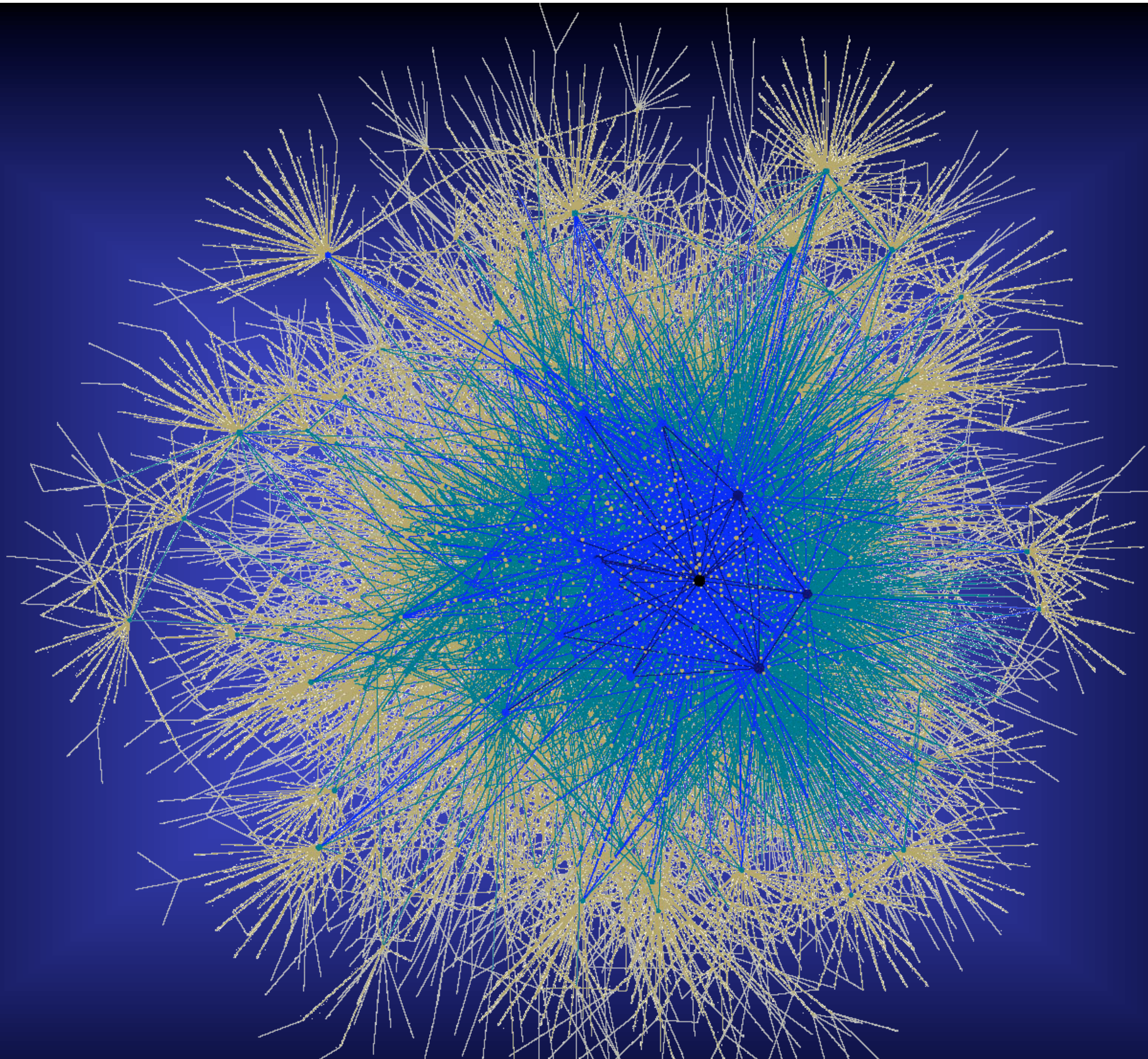
Information network



An induced subgraph of the collaboration graph with authors of Erdös number ≤ 2.

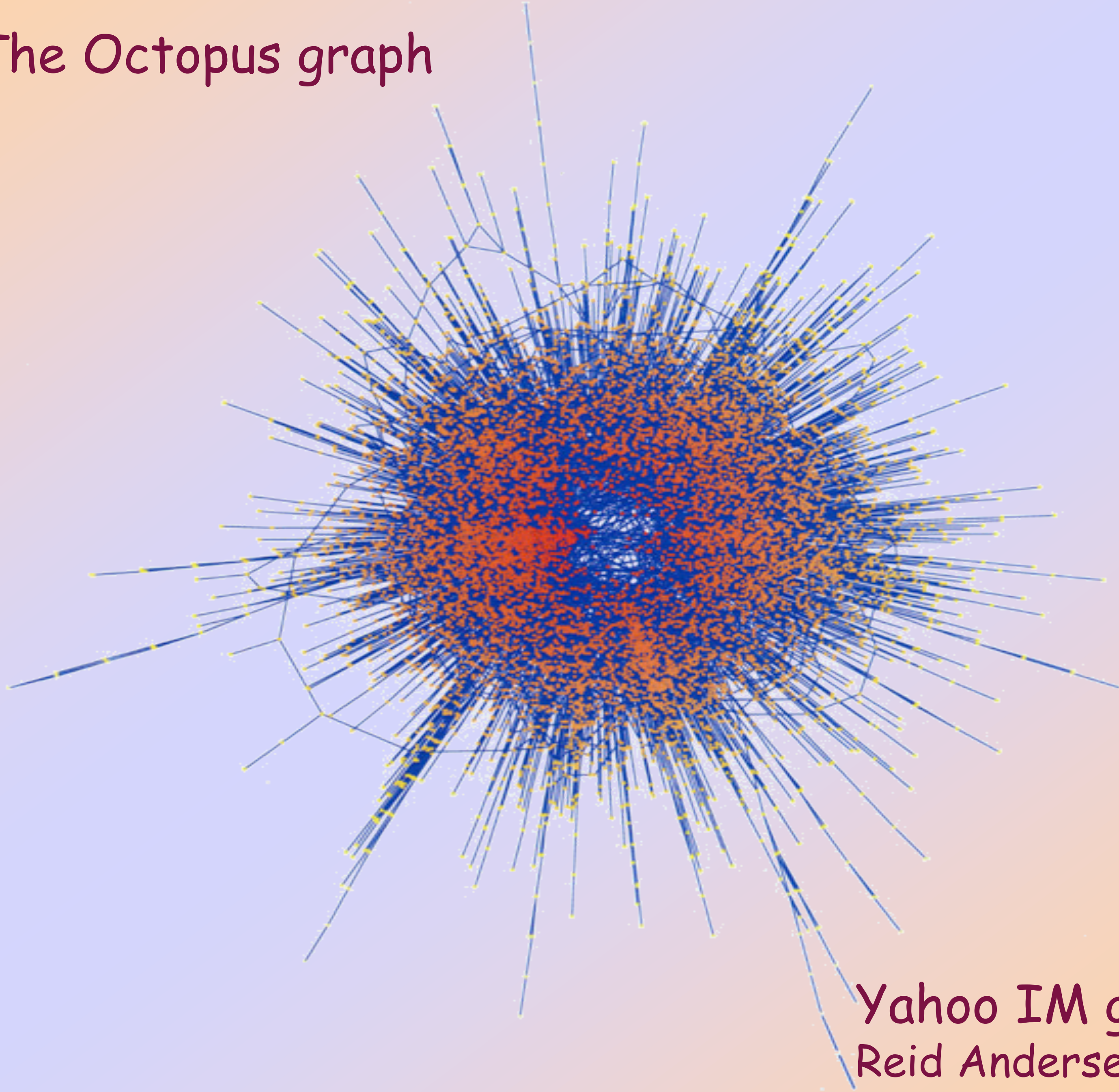


A subgraph of the Hollywood graph.



A subgraph of a BGP graph

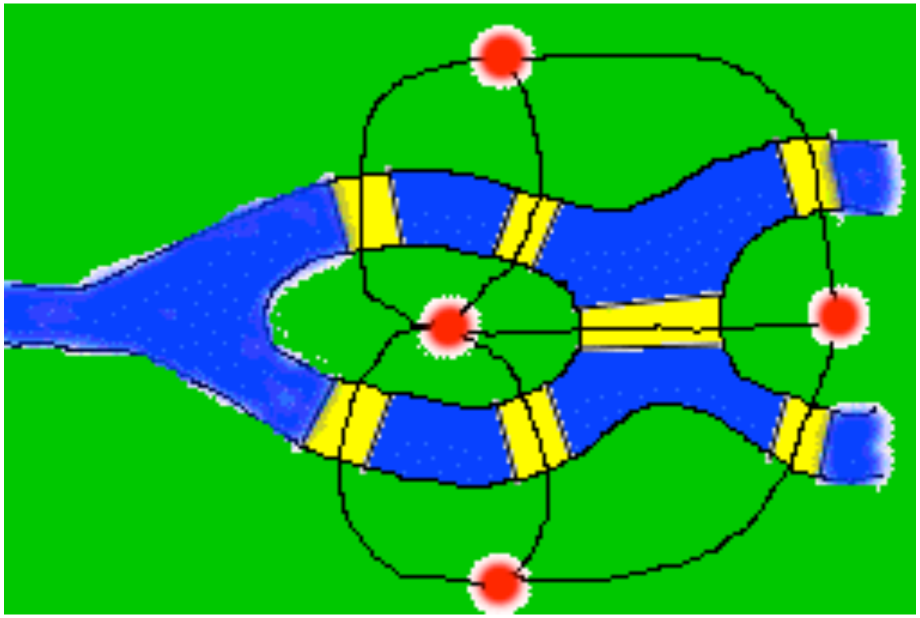
The Octopus graph



Yahoo IM graph
Reid Andersen

Graph Theory has 250 years of history.

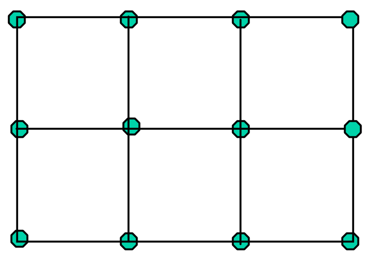
Leonhard Euler
1707-1783

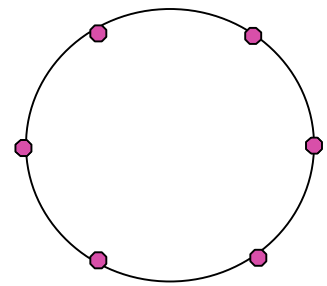


The Bridges of Königsburg

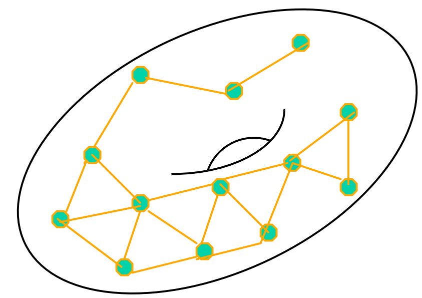
Is it possible to walk over every bridge once and only once?

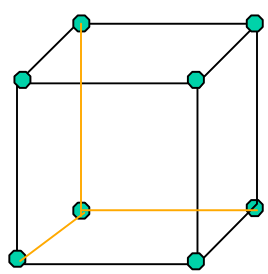
Geometric graphs

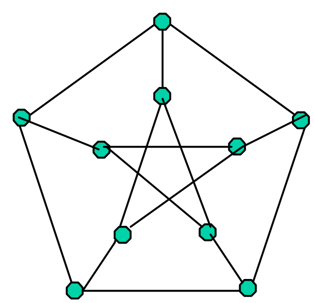




Topological graphs

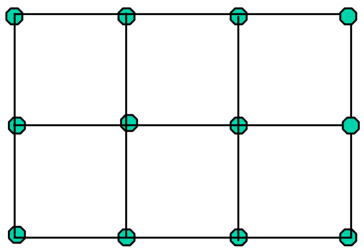


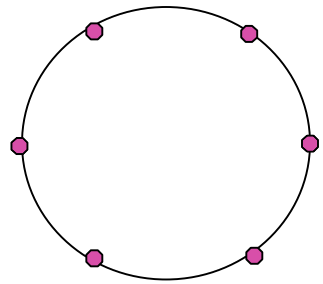




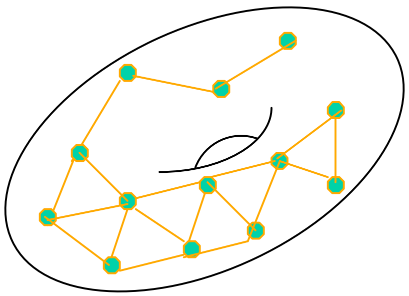
Algebraic graphs

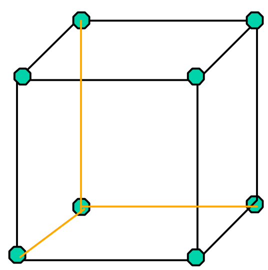
Geometric graphs

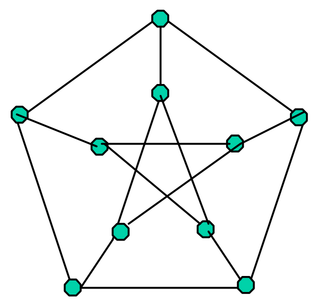




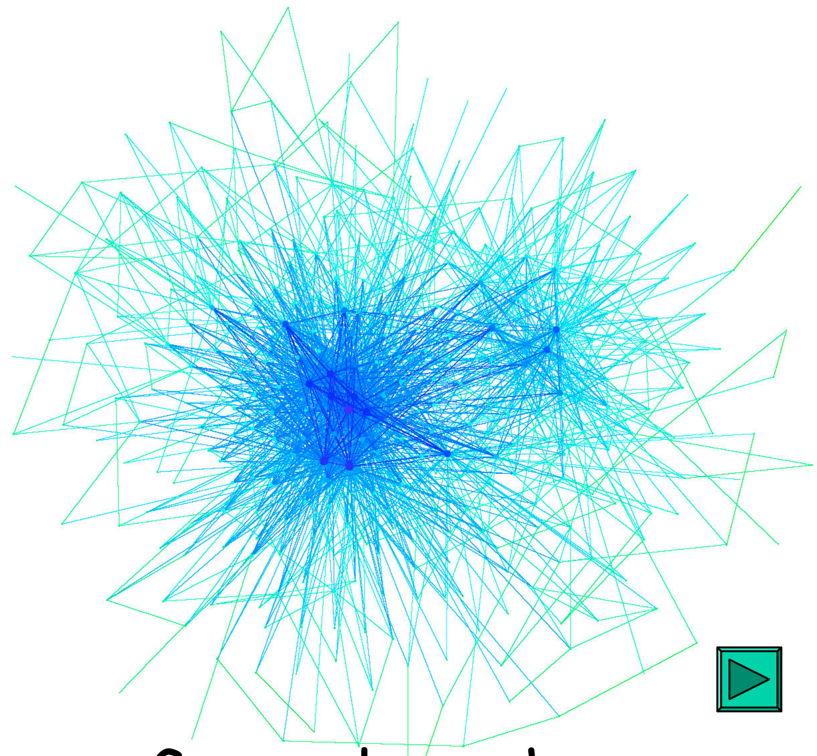
Topological graphs





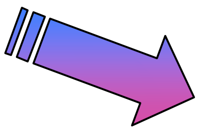


Algebraic graphs



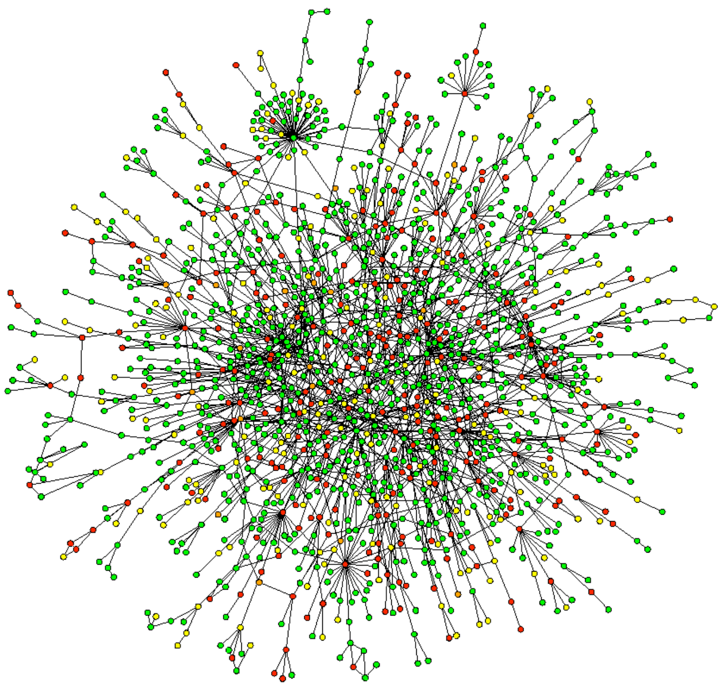
General graphs

Massive data

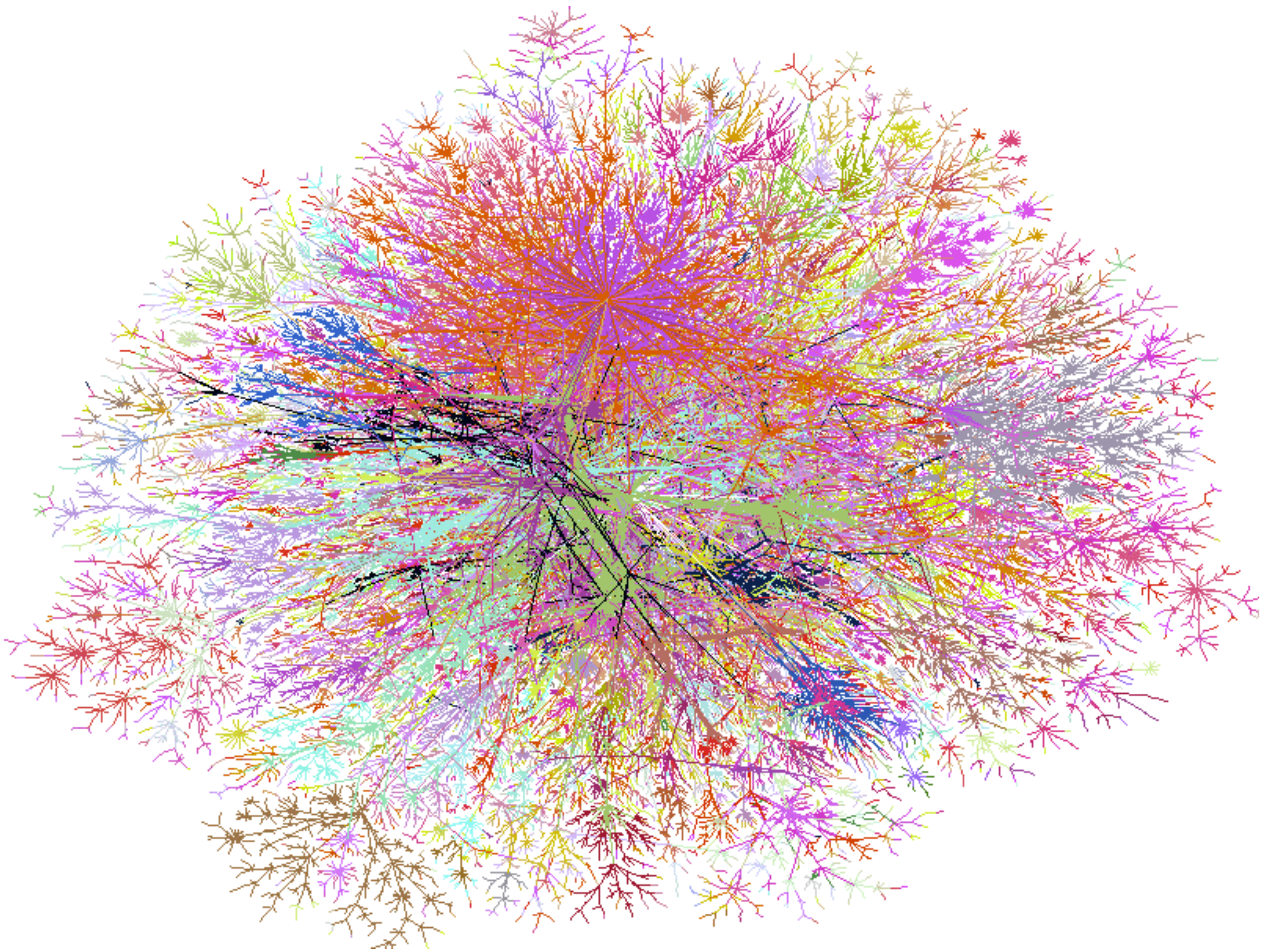


Massive graphs

- WWW-graphs
- Call graphs
- Acquaintance graphs
- Graphs from any data base



protein interaction network
Jawoong Jeong



Big and bigger graphs ⇨ New directions.

Efficient algorithms for massive networks

Basic questions:

- Correlation among nodes?
- The `geometry' of a network ?

  distance, flow, cut, …
- Quantitative analysis?

  eigenvalues, rapid mixing, …
- Local versus global?

Google's answer:

PageRank Explained

PageRank relies on the uniquely democratic nature of the web by using its vast link structure as an indicator of an individual page's value. In essence, Google interprets a link from page A to page B as a vote, by page A, for page B. But, Google looks at considerably more than the sheer volume of votes, or links a page receives; for example, it also analyzes the page that casts the vote. Votes cast by pages that are themselves "important" weigh more heavily and help to make other pages "important." Using these and other factors, Google provides its views on pages' relative importance.

The definition for PageRank?

A measure for the "importance" of a website

$$x_1 = R(x_{14} + x_{79} + x_{785})$$

$$x_2 = R(x_{1002} + x_{3225} + x_{9883} + x_{30027})$$

$$\cdots = \cdots$$

The "importance" of a website is proportional to the sum of the importance of all the sites that link to it.

Adjacency matrix of a graph

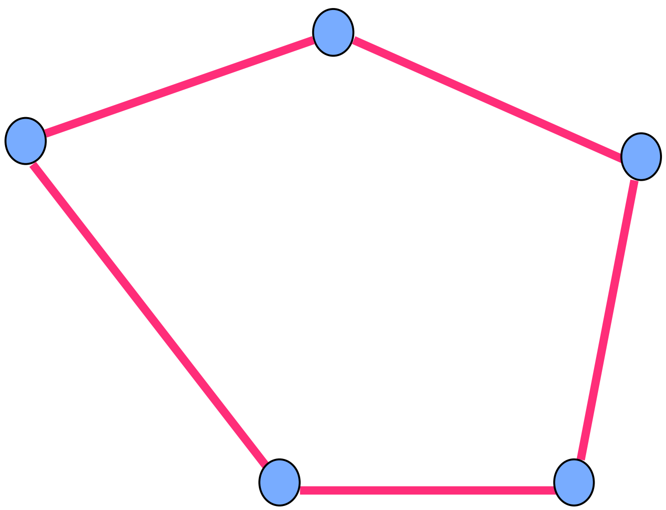
G : a graph on n vertices

A : adjacency matrix of G of size n x n

$$A(u,v) = \begin{cases} 1 & if\ u \sim v, \\ 0 & otherwise. \end{cases}$$

Adjacency matrix of a graph

Example: Adjacency matrix of a 5-cycle



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

A solution for the "importance" of a website

$$x_1 = \rho(x_{14} + x_{79} + x_{785})$$

$$x_2 = \rho(x_{1002} + x_{3225} + x_{9883} + x_{30027})$$

$$\cdots = \cdots$$

Solve $\quad x_i = \rho \sum_{j=1}^{n} a_{ij} x_j \quad$ for $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$

A solution for the "importance" of a website

$$x_1 = \rho(x_{14} + x_{79} + x_{785})$$

$$x_2 = \rho(x_{1002} + x_{3225} + x_{9883} + x_{30027})$$

$$\cdots = \cdots$$

Solve $x_i = \rho \sum_{j=1}^{n} a_{ij} x_j$ for $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$

$$\boldsymbol{x} = \rho A \boldsymbol{x} \qquad A = \left[a_{ij} \right]_{n \times n}$$

A solution for the "importance" of a website

$$x_1 = \rho(x_{14} + x_{79} + x_{785})$$

$$x_2 = \rho(x_{1002} + x_{3225} + x_{9883} + x_{30027})$$

$$\cdots = \cdots$$

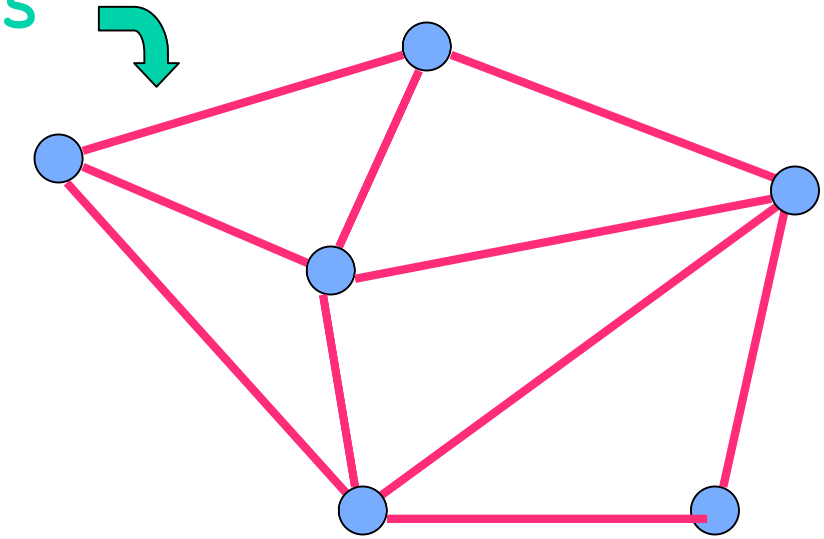
Solve $\quad x_i = \rho \sum_{j=1}^{n} a_{ij} x_j \quad$ for $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$

$$\boldsymbol{x} = \rho A \boldsymbol{x}$$

Eigenvalue problems!

Graph models

(undirected) graphs



directed graphs

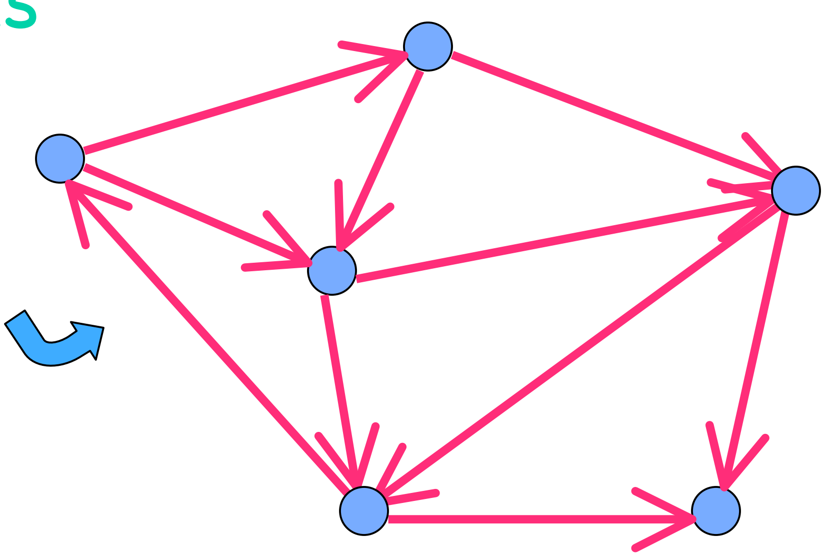
weighted graphs

Graph models

(undirected) graphs

directed graphs

weighted graphs

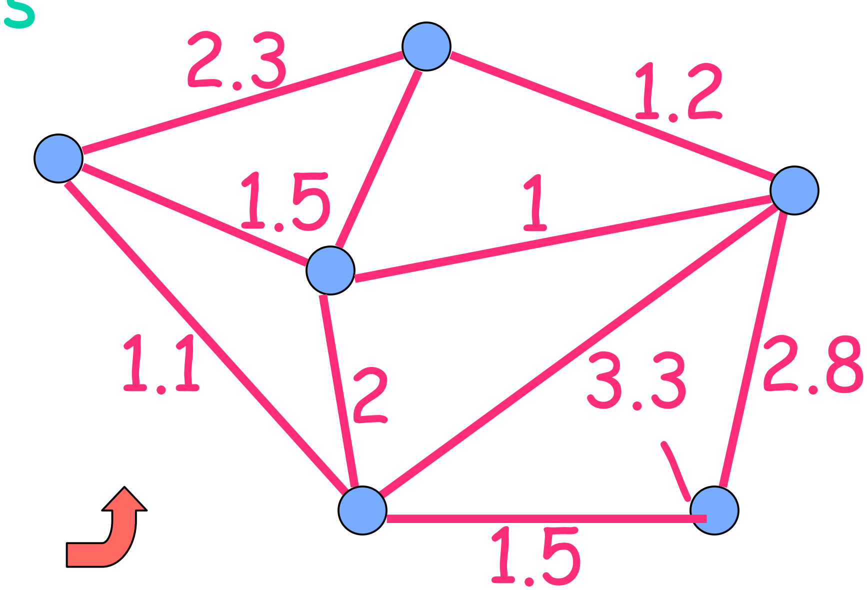


Graph models

(undirected) graphs

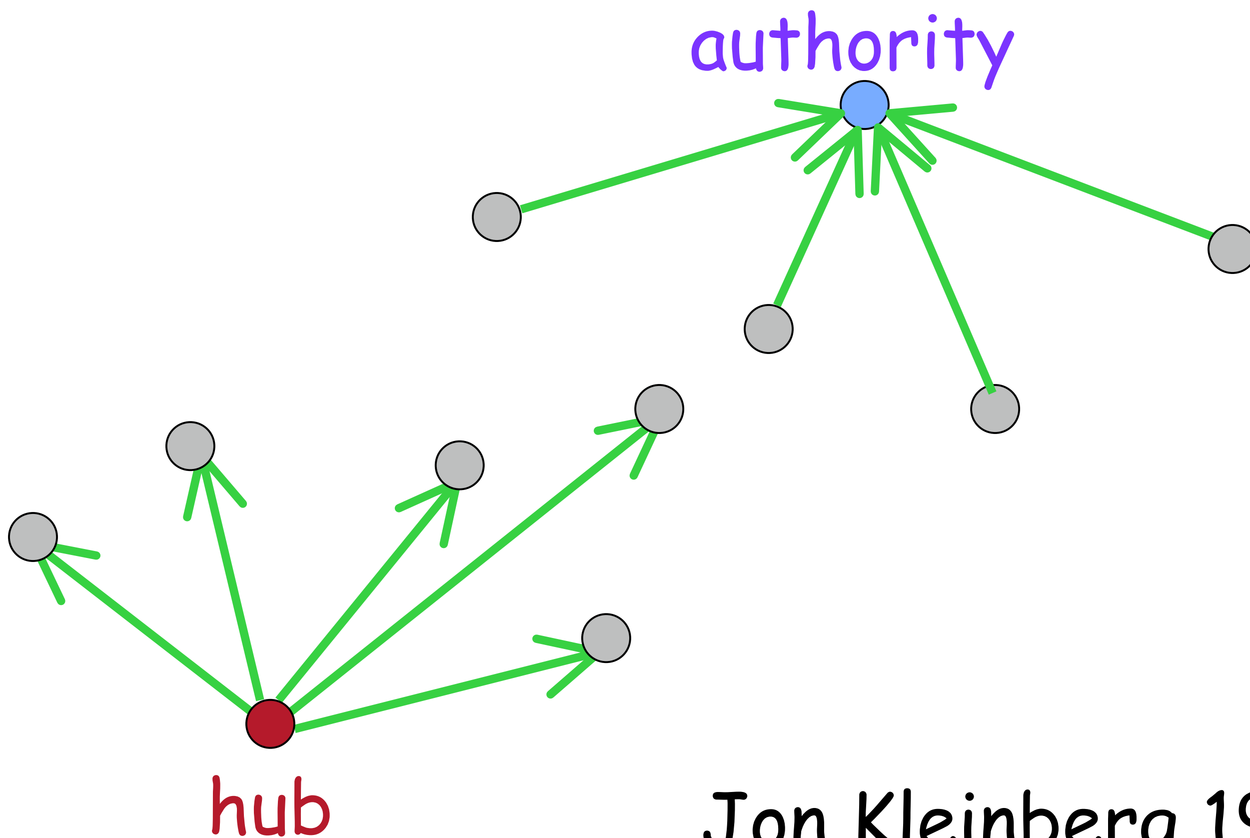
directed graphs

weighted graphs



In a directed graph,

there are two types of "importance":



Jon Kleinberg 1998

Two types of the "importance" of a website

Importance as Authorities : $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$

Importance as Hubs : $\boldsymbol{y} = (y_1, y_2, \cdots, y_m)$

Solve $\boldsymbol{x} = r\,A\,\boldsymbol{y}$ and $\boldsymbol{y} = s\,A^T\boldsymbol{x}$

$$\Rightarrow \begin{cases} \boldsymbol{x} = rs\,A\,A^T\boldsymbol{x} \\ \boldsymbol{y} = rs\,A^T A\,\boldsymbol{y} \end{cases}$$

Singular eigenvalue problems!

Eigenvalue problem for n x n matrix:.

$n \approx$ 30 billion websites

Hard to compute eigenvalues

Even harder to compute eigenvectors

In the old days,

compute for a given (whole) graph.

In reality,

can only afford to compute "locally".

(Access to a (huge) graph,

e.g., for a vertex v, find its neighbors.

Bounded number of access.)

A traditional algorithm

Input: a given graph on n vertices.

Efficient algorithm means polynomial algorithms

n^3, n^2, $n \log n$, n

New algorithmic paradiam

Input: access to a (huge) graph

(e.g., for a vertex v, find its neighbors)

Bounded number of access.

A traditional algorithm

Input: a given graph on n vertices.

Exponential

Efficient algorithm means polynomial algorithms

polynomial

n^3, n^2, n log n, n

New algorithmic paradigm

Input: access to a (huge) graph

Infinity

(e.g., for a vertex v, find its neighbors)

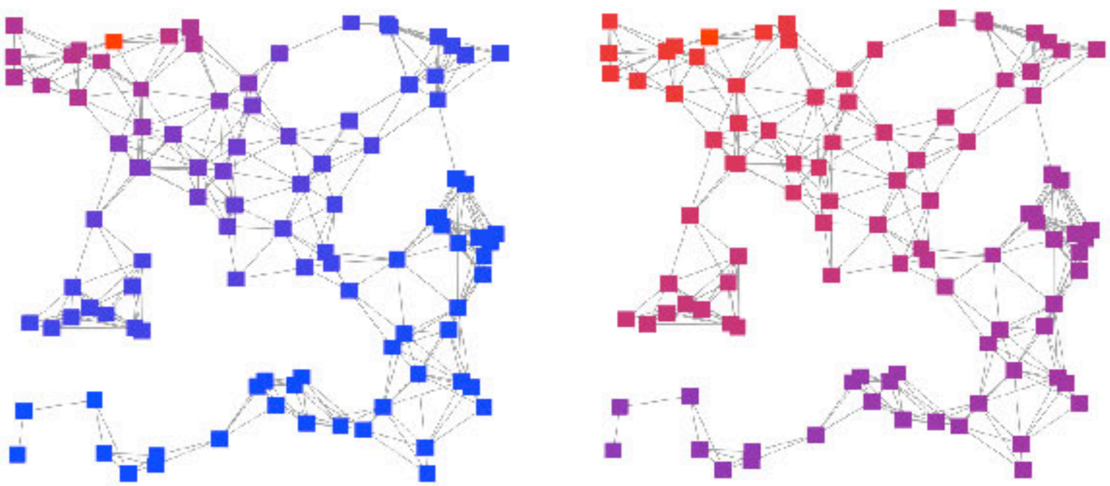
finite

Bounded number of access.

The definition of PageRank given by

Brin and Page is based on

random walks.



Random walks in a graph.

G : a graph

P : transition probability matrix

$$P(u,v) = \begin{cases} \dfrac{1}{d_u} & if\ u \sim v, \\ 0 & otherwise. \end{cases}$$

d_u := the degree of u.

A lazy walk: $$W = \frac{I+P}{2}$$

Original definition of PageRank

A (bored) surfer

- either surf a random webpage

  with probability α

- or surf a linked webpage

  with probability $1- \alpha$

α : the jumping constant

$$p = \alpha(\tfrac{1}{n}, \tfrac{1}{n}, \ldots, \tfrac{1}{n}) + (1-\alpha) pW$$

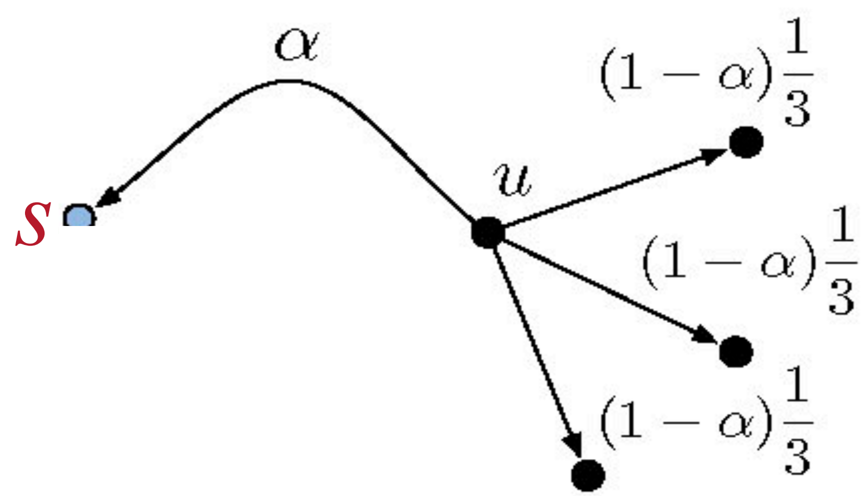
Definition of personalized PageRank

Two equivalent ways to define PageRank $pr(\alpha,s)$

(1) $$p = \alpha s + (1-\alpha) pW$$

s: the seed as a row vector

α : the jumping constant



Definition of PageRank

Two equivalent ways to define PageRank $p=pr(\alpha,s)$

(1) $$p = \alpha s + (1-\alpha) pW$$

(2) $$p = \alpha \sum_{t=0}^{\infty} (1-\alpha)^t (sW^t)$$

$s = (\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})$ ⟹ the (original) PageRank

$s =$ some "seed", e.g., $(1, 0, \ldots, 0)$

⟹ personalized PageRank

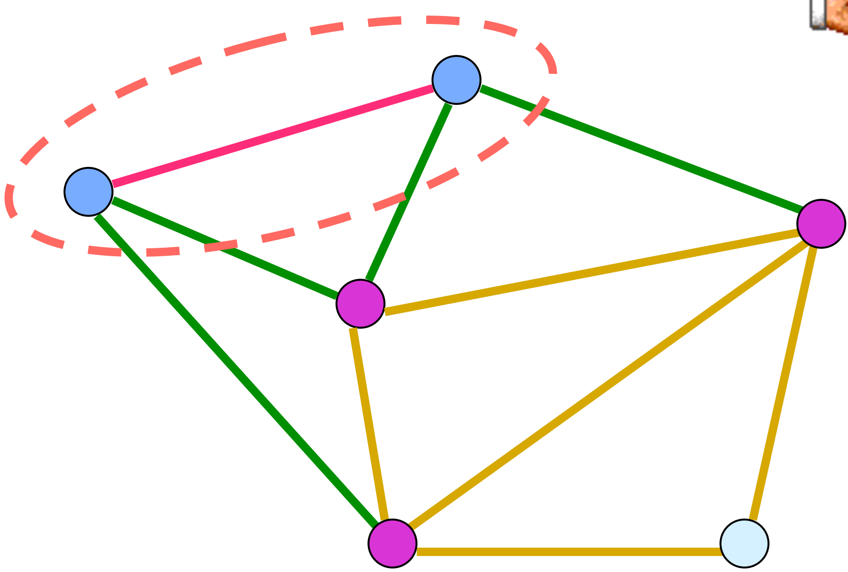
How good is a cut?

How good is PageRank for finding a good cut?

How "good" is the cut?

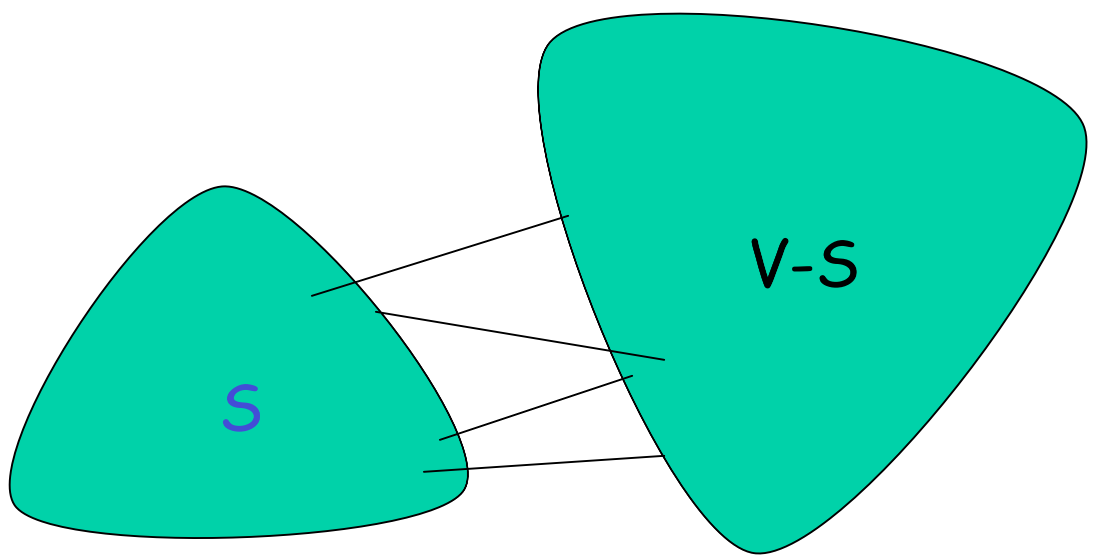
Two types of cuts:

- Vertex cut
- edge cut



$$\frac{e(S,V-S)}{\mathrm{Vol}\ S} \quad \longleftrightarrow \quad \frac{e(S,V-S)}{|S|}$$

$$\mathrm{Vol}\ S = \sum_{v \in S} \deg(v) \qquad |S| = \sum_{v \in S} 1$$



The Cheeger constant for graphs

The Cheeger constant

$$\Phi_G = \min_S \frac{e(S,\bar{S})}{\min(vol\ S, vol\ \bar{S})}$$

The volume of S is $$vol(S) = \sum_{x \in S} d_x$$

Φ_G and its variations are sometimes called "conductance", "isoperimetric number", ...

The Cheeger inequality

The Cheeger constant

$$\Phi_G = \min_S \frac{e(S,\bar{S})}{\min(vol\ S, vol\ \bar{S})}$$

The Cheeger inequality

$$2\Phi_G \geq \lambda \geq \frac{\Phi_G^{\,2}}{2}$$

λ : the first nontrivial eigenvalue of the (normalized) Laplacian.

The spectrum of a graph

- Adjacency matrix

Many ways to define the spectrum of a graph.

How are the eigenvalues related to properties of graphs?

The spectrum of a graph

- Adjacency matrix

- Combinatorial Laplacian

$$L = D - A$$

adjacency matrix

diagonal degree matrix

- Normalized Laplacian

Random walks

Rate of convergence

The spectrum of a graph

loopless, simple

Discrete Laplace operator

$$\Delta f(x) = \frac{1}{d_x}\sum_{y\sim x}(f(x)-f(y))$$

$$L(x,y) = \begin{cases} 1 & if \quad x = y \\ -\dfrac{1}{d_x} & if \quad x \neq y \quad and \quad x \sim y \end{cases}$$

not symmetric in general

• Normalized Laplacian

symmetric
normalized

$$\mathrm{L}(x,y) = \begin{cases} 1 & if \quad x = y \\ -\dfrac{1}{\sqrt{d_x d_y}} & if \quad x \neq y \quad and \quad x \sim y \end{cases}$$

with eigenvalues

$$0 = \lambda_0 \le \lambda_1 \le \cdots \le \lambda_{n-1} \le 2$$

The spectrum of a graph

Discrete Laplace operator

$$\Delta f(x) = \frac{1}{d_x} \sum_{y \sim x} (f(x) - f(y))$$

$$L(x,y) = \begin{cases} 1 & if \;\; x = y \\ -\dfrac{1}{d_x} & if \;\; x \neq y \;\; and \;\; x \sim y \end{cases}$$

not symmetric in general

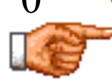
- Normalized Laplacian

symmetric
normalized

$$\mathrm{L}(x,y) = \begin{cases} 1 & if \;\; x = y \\ -\dfrac{1}{\sqrt{d_x d_y}} & if \;\; x \neq y \;\; and \;\; x \sim y \end{cases}$$

with eigenvalues

$$0 = \lambda_0 \leq \underset{\lambda}{\underset{\shortparallel}{\lambda_1}} \leq \cdots \leq \lambda_{n-1} \leq 2$$



Can you hear the shape of a network?

λ dictates many properties of a graph.

- connectivity
- diameter
- isoperimetry (bottlenecks)
- ... ...

How "good" is the cut by using the eigenvalue λ ?

Finding a cut by a sweep

Using a sweep by the eigenvector,

can reduce the exponential number of

choices of subsets to a **linear** number.

Finding a cut by a sweep

Using a sweep by the eigenvector, can reduce the exponential number of choices of subsets to a linear number.

- Still, there is a lower bound guarantee by using the Cheeger inequality.

$$2\Phi \geq \lambda \geq \frac{\Phi^2}{2}$$

Four one-sweep graph partitioning algorithms

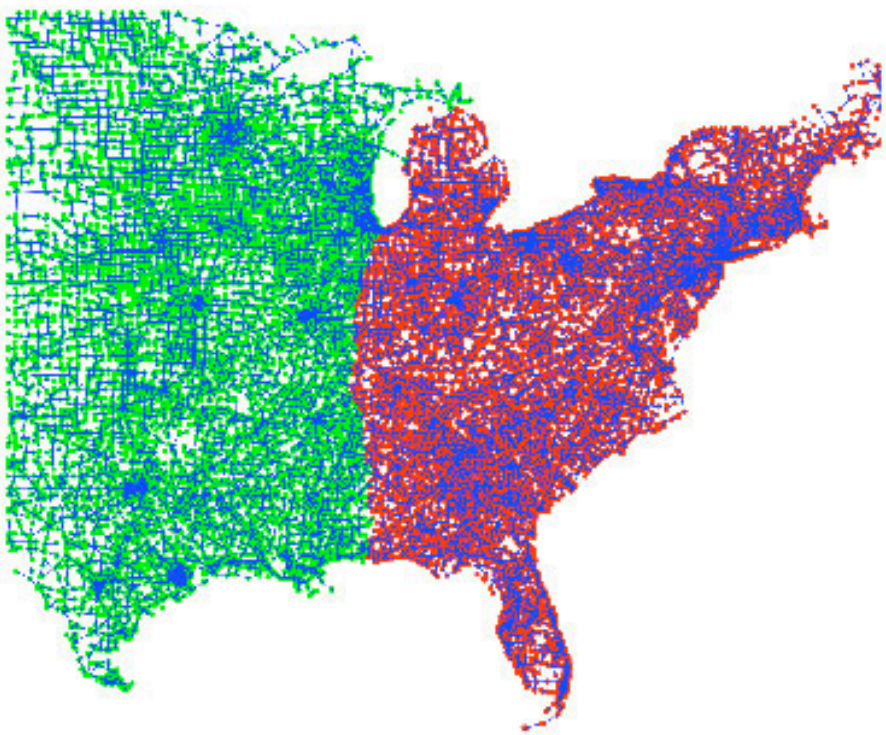
- graph spectral method

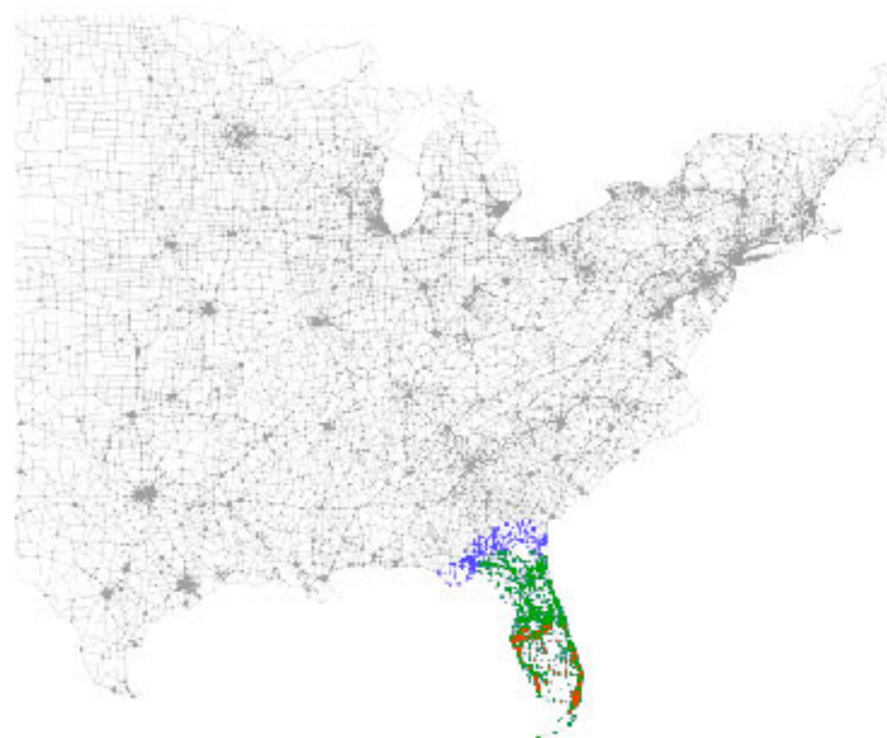
spectral partition algorithm

- random walks
- PageRank
- heat kernel

local partition algorithms

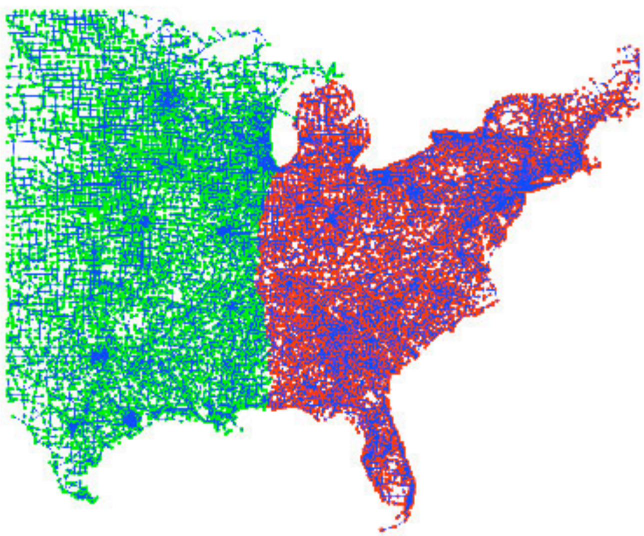
Graph partitioning ⟷ Local graph partitioning

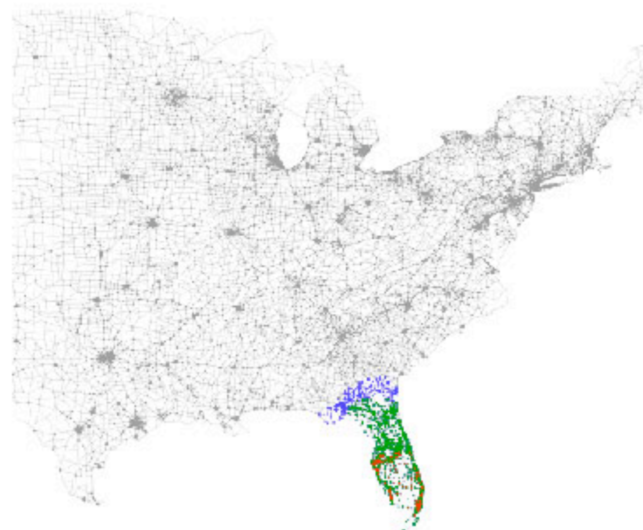


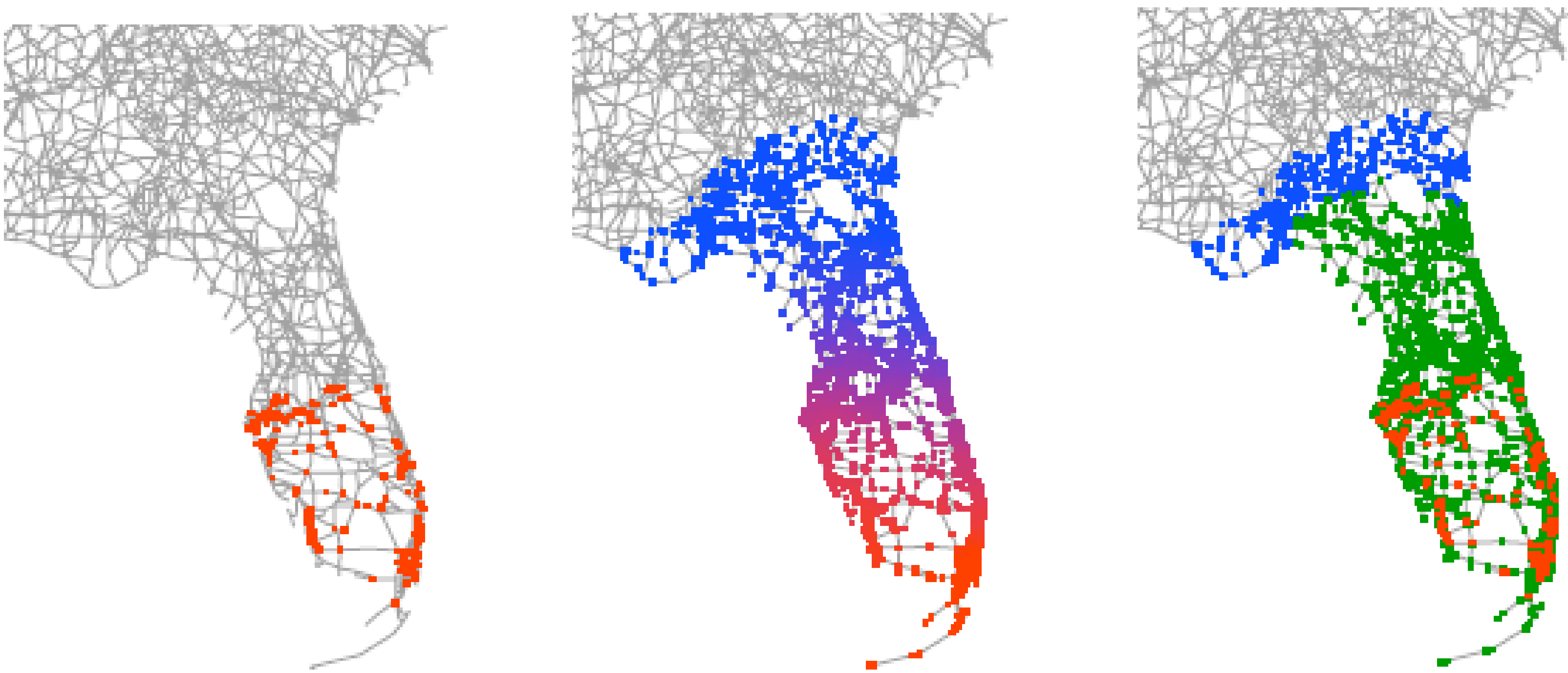


What is a local graph partitioning algorithm?

A local graph partitioning algorithm finds a small cut near the given seed(s) with running time depending only on the size of the output.







Challenges

Finding isolated submarkets is a difficult partitioning problem.
(sparsest cut problem, minimum conductance cut problem).

The graph can be prohibitively large.

- It might not fit in memory.
- You might have only streaming access to the edges.
- You might have access to the graph over a network.

The best solutions can be very small.

Small solutions

Q. What is the single most isolated submarket in the graph?

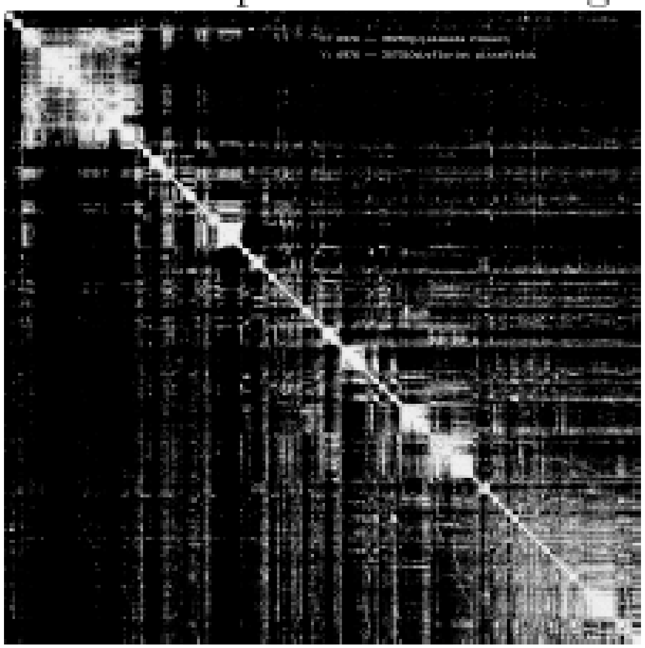
A. It is probably the set containing 6 advertisers and 26 phrases about ferry boats, with 1 bid leaving the submarket.

`brittany ferry, calais dover ferry, ferry spain, channel cross ferry, channel english ferry, england ferry france,`

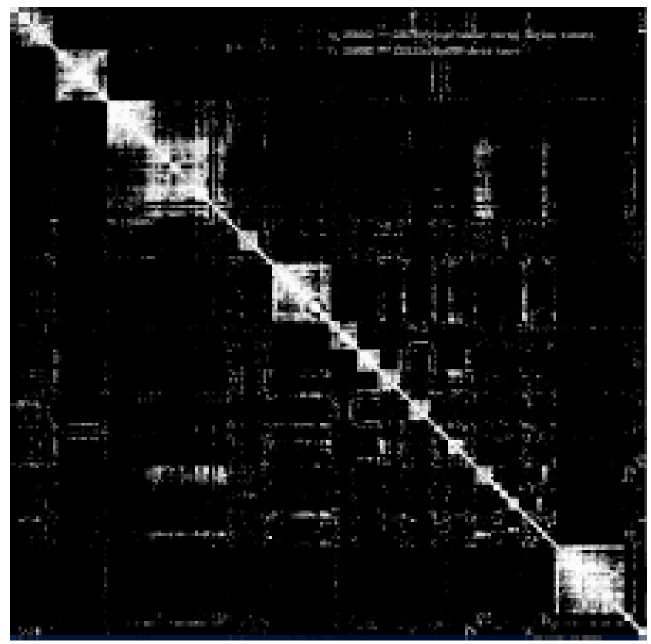
This was the answer obtained by performing spectral partitioning on the entire 2 million-edge graph.

There are thousands of submarkets

Full sponsored search graph



10x zoom



When is local partitioning useful?

1. Finding a small community in a large graph.

Example in the sponsored search graph.

Starting with the seed vertex "alameda flower", our algorithm finds a set of 300 bidders and phrases related to flower stores in the San Francisco area. Few bids leave this isolated submarket.

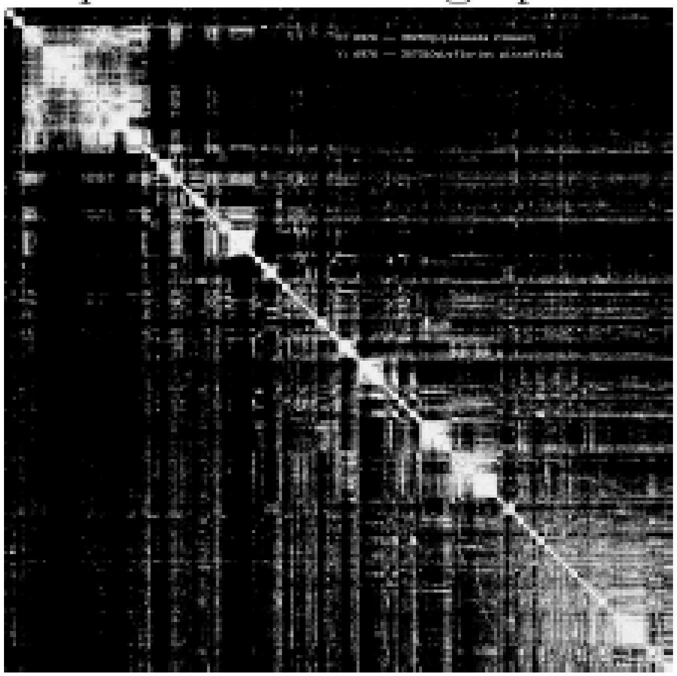
```
alameda flower,      florist francisco in san,
alameda florist,     flower menlo park,
burlingame flower,   bruno flower san,
flower rafael san,   city flower redwood,
city daly flower,    florist rafael san,    . . .
```

To find this submarket, our algorithm examined only 1200 vertices out of $653,260$ in the graph.

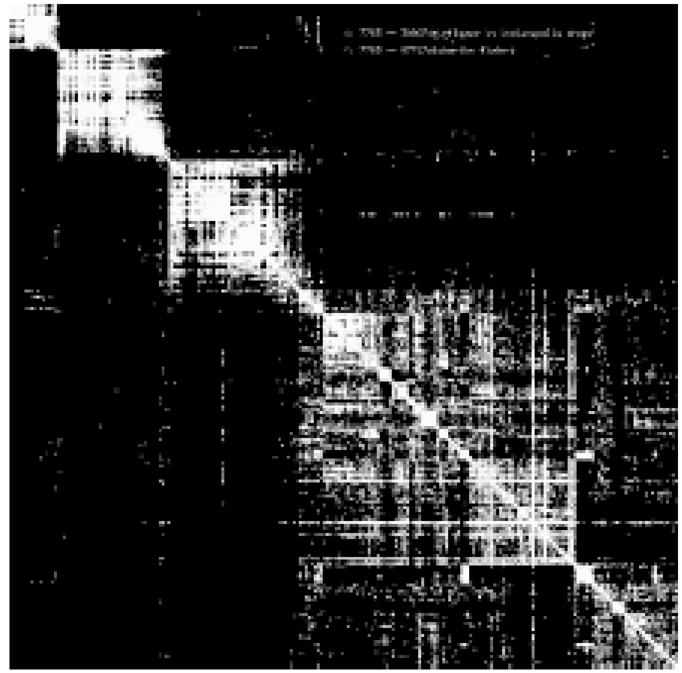
When is local partitioning useful?

2. Finding every community in a large graph.

Sponsored search graph



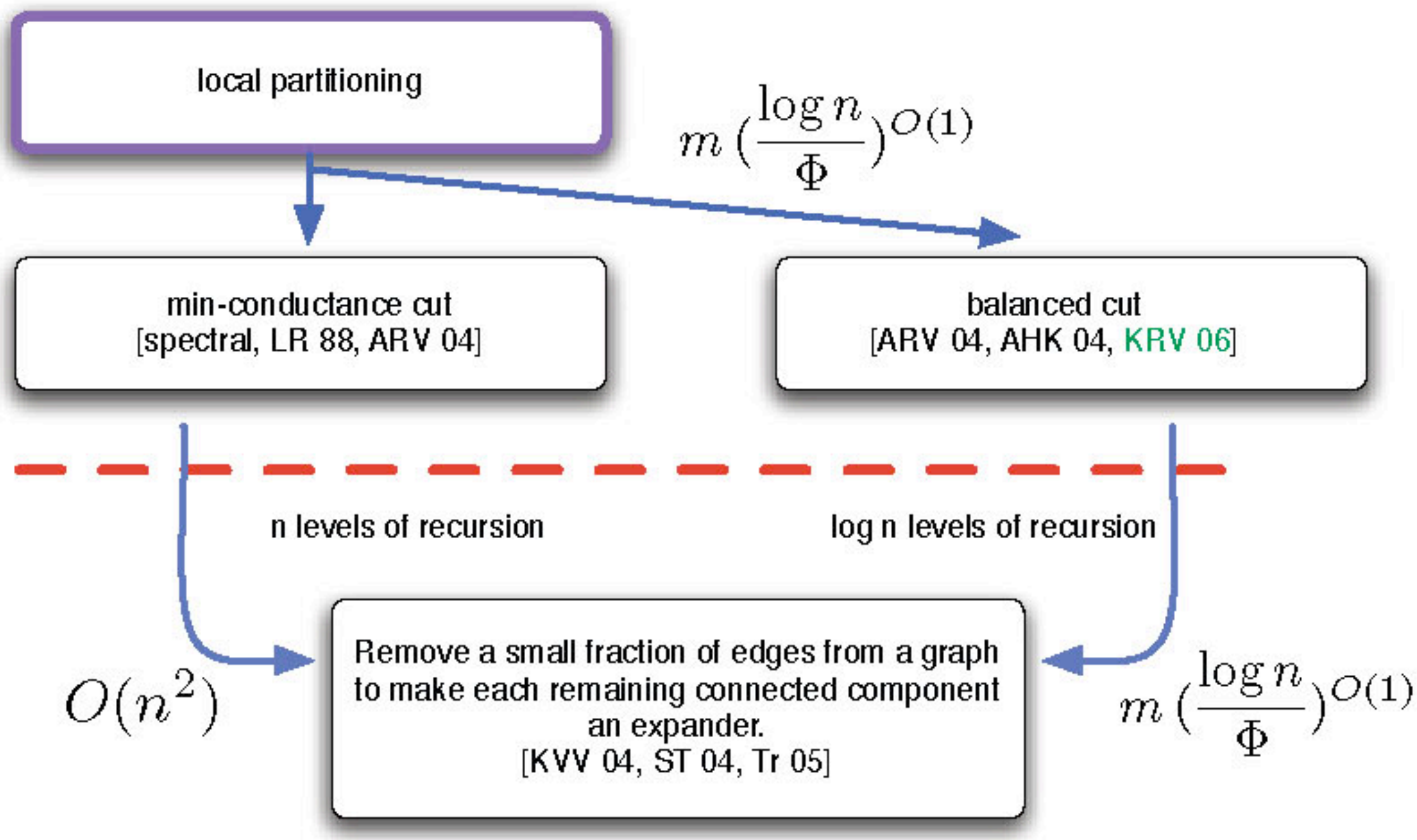
Zoomed in on the flowers



By picking many random seed vertices and target sizes, we can cover the graph with numerous submarkets. The time required is roughly the same as computing PageRank with the power method $\log n$ times.

Theoretical applications

Local partitioning can be used as a subroutine to more quickly solve problems traditionally solved by recursive partitioning.



Four one-sweep graph partitioning algorithms

- graph spectral method
- random walks
- PageRank
- heat kernel

α

Partitioning algorithm ⟷ The Cheeger inequality

Using eigenvector f,

the Cheeger inequality can be stated as

$$2\Phi \geq \lambda \geq \frac{\alpha^2}{2} \geq \frac{\Phi^2}{2}$$

where λ is the first non-trivial eigenvalue of the Laplacian and α is the minimum Cheeger ratio in a sweep using the eigenvector f.

Proof of the Cheeger inequality:

$$
\begin{aligned}
\lambda_G &\geq R(g_+) \\
&= \frac{\sum_{u\sim v}(g_+(u)-g_+(v))^2}{\sum_u g_+^2(u)d_u} \\
&= \frac{\left(\sum_{u\sim v}(g_+(u)-g_+(v))^2\right)\left(\sum_{u\sim v}(g_+(u)+g_+(v))^2\right)}{\sum_u g_+^2(u)d_u \sum_{u\sim v}(g_+(u)+g_+(v))^2} \\
&\geq \frac{\left(\sum_{u\sim v}(g_+(u)^2-g_+(v)^2)\right)^2}{2\left(\sum_u g_+^2(u)d_u\right)^2} \\
&= \frac{\left(\sum_i |g_+(v_i)^2-g_+(v_{i+1})^2| \quad |\partial(S_i)|\right)^2}{2\left(\sum_u g_+^2(u)d_u\right)^2} \\
&\geq \frac{\left(\sum_i |g_+(v_i)^2-g_+(v_{i+1})^2|\alpha_G|\tilde{\mathrm{vol}}(S_i)|\right)^2}{2\left(\sum_u g_+^2(u)d_u\right)^2} \\
&= \frac{\alpha_G^2}{2}\frac{\left(\sum_i g_+(v_i)^2(|\tilde{\mathrm{vol}}(S_i)-\tilde{\mathrm{vol}}(S_{i+1})|)\right)^2}{\left(\sum_u g_+^2(u)d_u\right)^2} \\
&= \frac{\alpha_G^2}{2}\frac{\left(\sum_i g_+(v_i)^2 d_{v_i}\right)^2}{\left(\sum_u g_+^2(u)d_u\right)^2} \\
&= \frac{\alpha_G^2}{2}.
\end{aligned}
$$

from definition

by Cauchy-Schwarz ineq.

from the definition.

summation by parts.

The Cheeger inequality $\Longleftrightarrow$ Partition algorithm

Using the eigenvector f,

the Cheeger inequality can be stated as

$$2\Phi \geq \lambda \geq \frac{\alpha^2}{2} \geq \frac{\Phi^2}{2}$$

where λ is the first non-trivial eigenvalue of the Laplacian and α is the minimum Cheeger ratio in a sweep using the eigenvector f .

Four graph partitioning algorithms

- graph spectral method

spectral partition algorithm

- random walks
- PageRank
- heat kernel

local partition algorithms

4 Partitioning algorithm ⟷ 4 Cheeger inequalities

- graph spectral method — Fiedler '73, Cheeger, 60's
- random walks — Mihail 89; Lovasz, Simonovits, 90, 93; Spielman, Teng, 04
- PageRank — Andersen, Chung, Lang, 06
- heat kernel — Chung, PNAS , 08.

Partitioning algorithm using random walks

Mihail 89, Lovász+Simonovits, 90, 93

$$\left|W^k(u,S)-\pi(S)\right| \le \sqrt{\frac{vol(S)}{d_u}}\left(1-\frac{\beta_k^2}{8}\right)^k$$

Leads to a Cheeger inequality:

$$2\Phi \;\ge\; \lambda \;\ge\; \frac{\beta_G{}^2}{8\log n} \;\ge\; \frac{\Phi^2}{8\log n}$$

where β_G is the minimum Cheeger ratio over sweeps by using a lazy walk of k steps from every vertex for an appropriate range of k .

Partitioning algorithm using PageRank

Using the PageRank vector.

Recall the definition of PageRank $p=pr(\alpha,s)$:

(1) $$p = \alpha s + (1-\alpha) pW$$

(2) $$p = \alpha \sum_{t=0}^{\infty} (1-\alpha)^t (sW^t)$$

Organize the random walks by a scalar α.

Partitioning algorithm using PageRank

Using the PageRank vector with seed as a subset S

and $vol(S) \leq vol(G)/4$, a Cheeger inequality

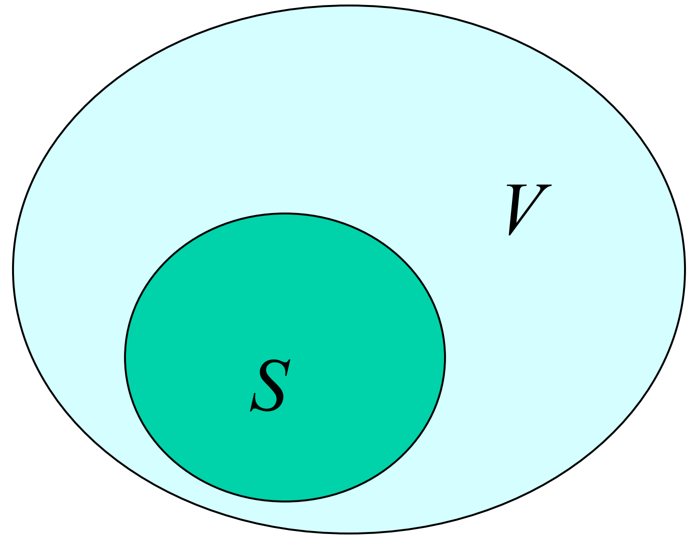
can be obtained :

$$\Phi_S \geq \lambda_S \geq \frac{{\gamma_S}^2}{8\log s} \geq \frac{{\Phi_S}^2}{8\log s}$$

where λ_S is the Dirichlet eigenvalue of the Laplacian, and γ_S is the minimum Cheeger ratio over sweeps by using personalized PageRank with seeds S.

Dirichlet eigenvalues for a subset $S \subseteq V$

$$\lambda_S = \inf_f \frac{\sum_{u \sim v} (f(u) - f(v))^2}{\sum_w f(w)^2 d_w}$$



over all f satisfying the Dirichlet boundary condition:

$f(v) = 0$ for all $v \notin S$.

Partitioning algorithm using PageRank

Using the PageRank vector with seed as a subset S and $vol(S) \leq vol(G)/4$, a Cheeger inequality can be obtained :

$$\Phi_S \geq \frac{\gamma_u^{\,2}}{8 \log s} \geq \frac{\Phi_u^{\,2}}{8 \log s}$$

where γ_u is the minimum Cheeger ratio over sweeps by using personalized PageRank with a random seed in S. The volume of the set of such u is $>$ vol(S)/4.

Partitioning ⟵ Computing PageRank

History of computing Pagerank

- Brin+Page 98
- Personalized PageRank, Haveliwala 03
- Computing personalized PageRank,
  Jeh+Widom 03

Algorithmic aspects of PageRank

- Fast approximation algorithm for personalized PageRank

  greedy type algorithm, linear complexity

- Can use the jumping constant to approximate PageRank with a support of the desired size.

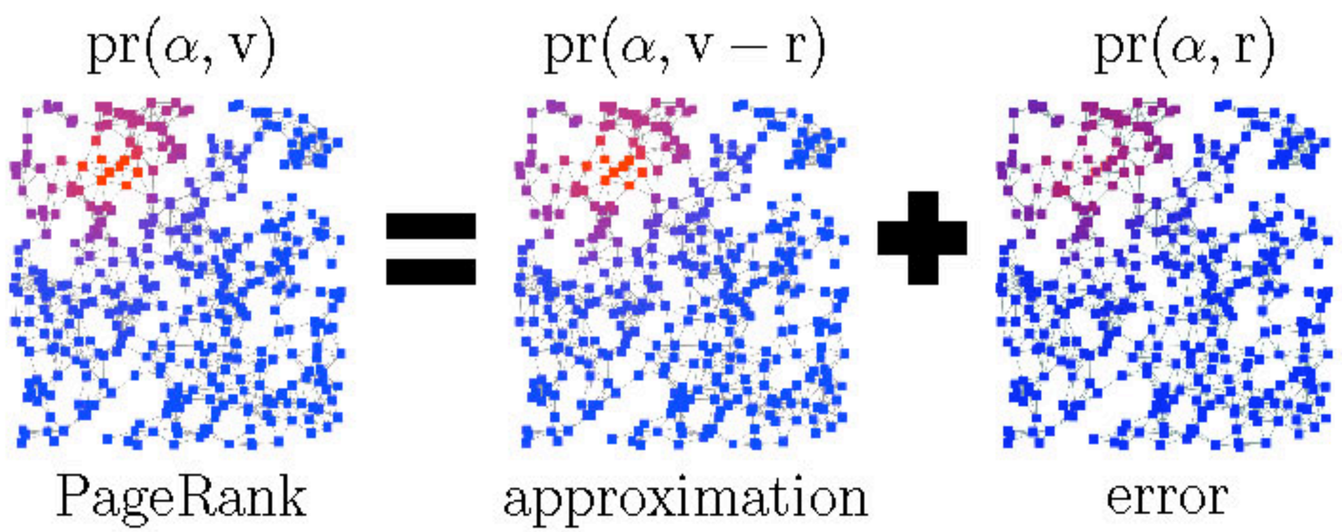
- Errors can be effectively bounded.

Approximate the pagerank vector :

$$pr(\alpha, s) = p + pr(\alpha, r)$$

Approximate pagerank

Residue vector



Computing PageRank

Lemma. For any $\epsilon > 0$, we can compute a PageRank vector $\mathrm{pr}(\alpha, v - r)$ where the error vector r satisfies

$$\max_u \frac{r(u)}{d(u)} \leq \epsilon.$$

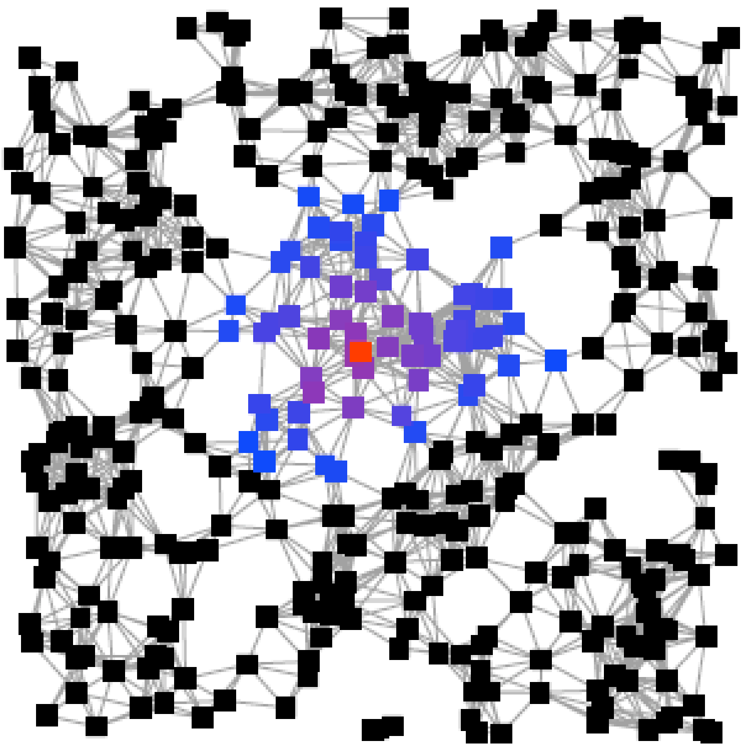
The time required is $\frac{1}{\epsilon\alpha}$. The set of vertices with nonzero probability from this approximation has volume at most $\frac{2}{\epsilon}$.

Sketch of algorithm.
Maintain an approximation p and a residual vector r which together satisfy the invariant

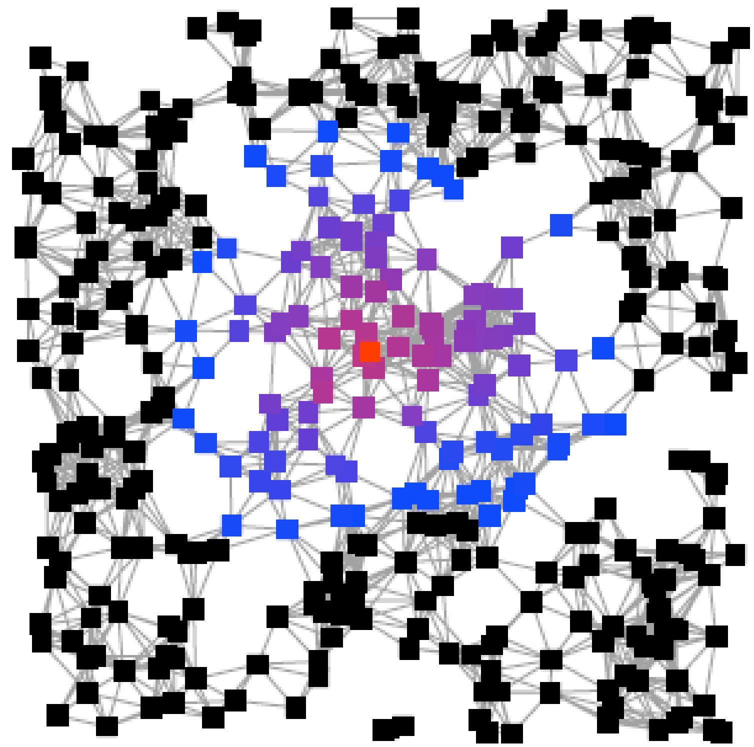
$$p + \mathrm{pr}(\alpha, r) = \mathrm{pr}(\alpha, v).$$

Initially, set $p = 0$ and set $r = v$.

Exploring a graph by computing PageRank



$\epsilon = .001$



$\epsilon = .0005$

$\alpha = .01$

Time spent on push operations:	$1/\alpha\epsilon$
Volume of examined vertices:	$1/\epsilon$

Partitioning algorithm using PageRank

Using the PageRank vector with seed as a subset S and $vol(S) \le vol(G)/4$, a Cheeger inequality can be obtained :

$$\Phi_S \geq \frac{\gamma_u^{\ 2}}{8\log s} \geq \frac{\Phi_u^{\ 2}}{8\log s}$$

where γ_u is the minimum Cheeger ratio over sweeps by using personalized PageRank with a random seed in S. The volume of the set of such u is $>$ vol(S)/4.

A partitioning algorithm using PageRank

Algorithm(φ,s,b):

- Compute ε-approximate Pagerank $p=pr(\alpha,s)$ with $\alpha=0.1/(\varphi^2\ b)$, $\varepsilon=2^{-b}/b$.
- One sweep algorithm using p for finding cuts with conductance $< \phi$.

Performance analysis:

If s is in a set S with conductance $\Phi>\varphi^2 log\ s$, with constant probability, the algorithm outputs a cut C with condutance $< \varphi$, of size order s and $vol(C \cap S) > \frac{1}{4} vol(S)$.

(Improving previous bounds by a factor of φlog s.)

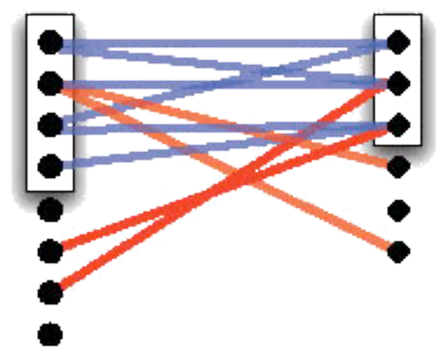
Comparison of local partitioning algorithms

Spielman and Teng 04:
A local partitioning algorithm with approximation quality
$\beta(\phi) = \phi^3 / \log^2 m$ that runs in time $x \cdot \left(\frac{\log^4 m}{\phi^5}\right)$.

Andersen, Chung, Lang 06:
A local partitioning algorithm with approximation quality
$\beta(\phi) = \phi^2 / \log^2 m$ that runs in time $x \cdot \left(\frac{\log^2 m}{\phi^2}\right)$.

Finding submarkets in the sponsored search graph

Task. Find sets of advertisers and phrases that form isolated submarkets, with few edges leaving the submarket.



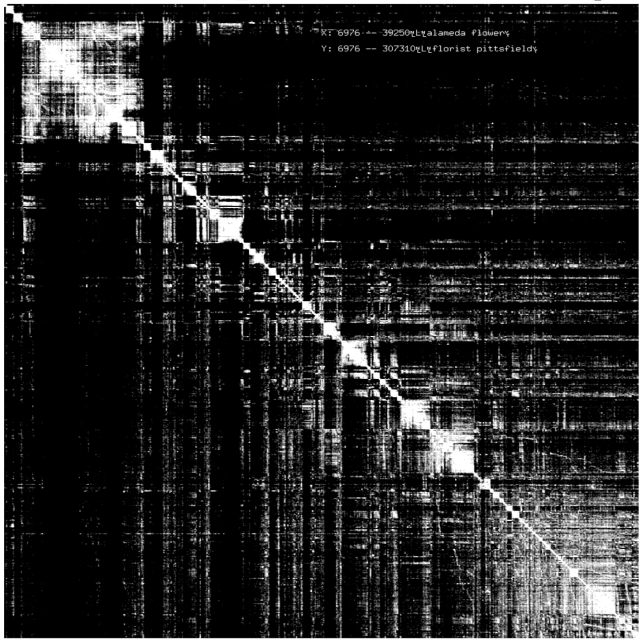
Applications

- Find groups of related phrases to suggest to advertisers.
- Find small submarkets for testing and experimentation.

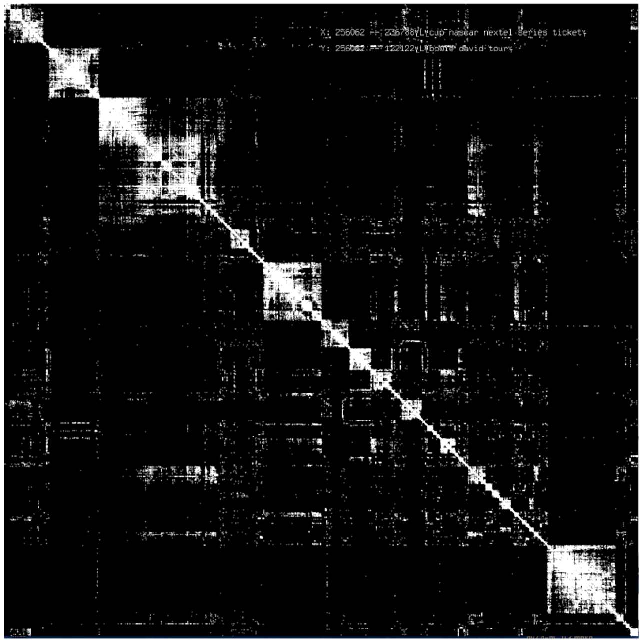
Courtesy of Reid Andersen.

There are thousands of submarkets

Full sponsored search graph



10x zoom



Courtesy of Reid Andersen

How do we use PageRank to partition?

Personalized PageRank [Brin/Page 98, Haveliwala 03] ranks vertices by their relevance to a given seed vertex.

The top 10 phrases most related to `alameda flower` according to a personalized PageRank vector.

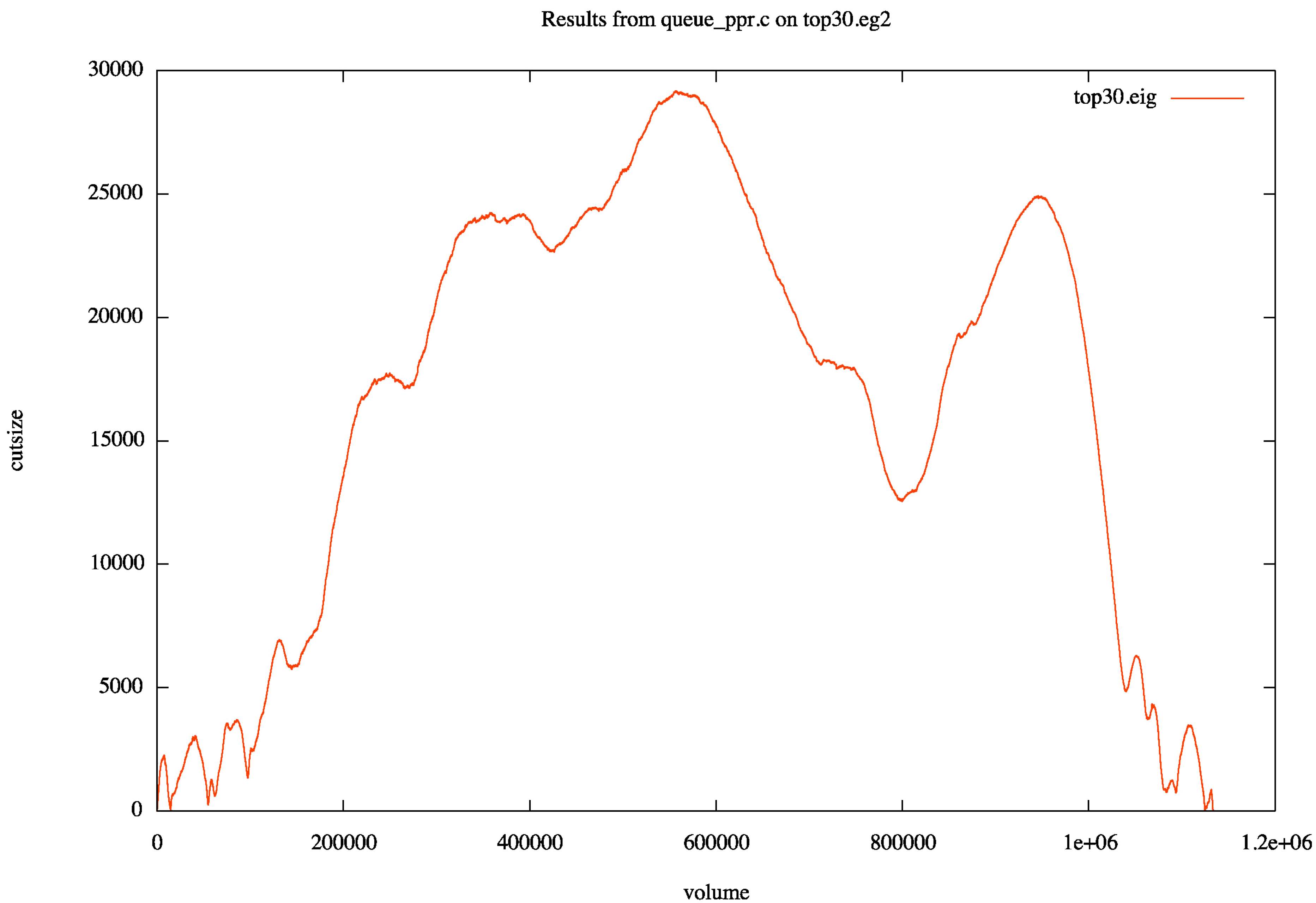
```
0  alameda flower
1  florist francisco in san
2  alameda florist
3  flower menlo park
4  burlingame flower
5  bruno flower san
6  flower rafael san
7  city flower redwood
8  city daly flower
9  florist rafael san
10 delivery flower francisco san
```

How do we use PageRank to partition?

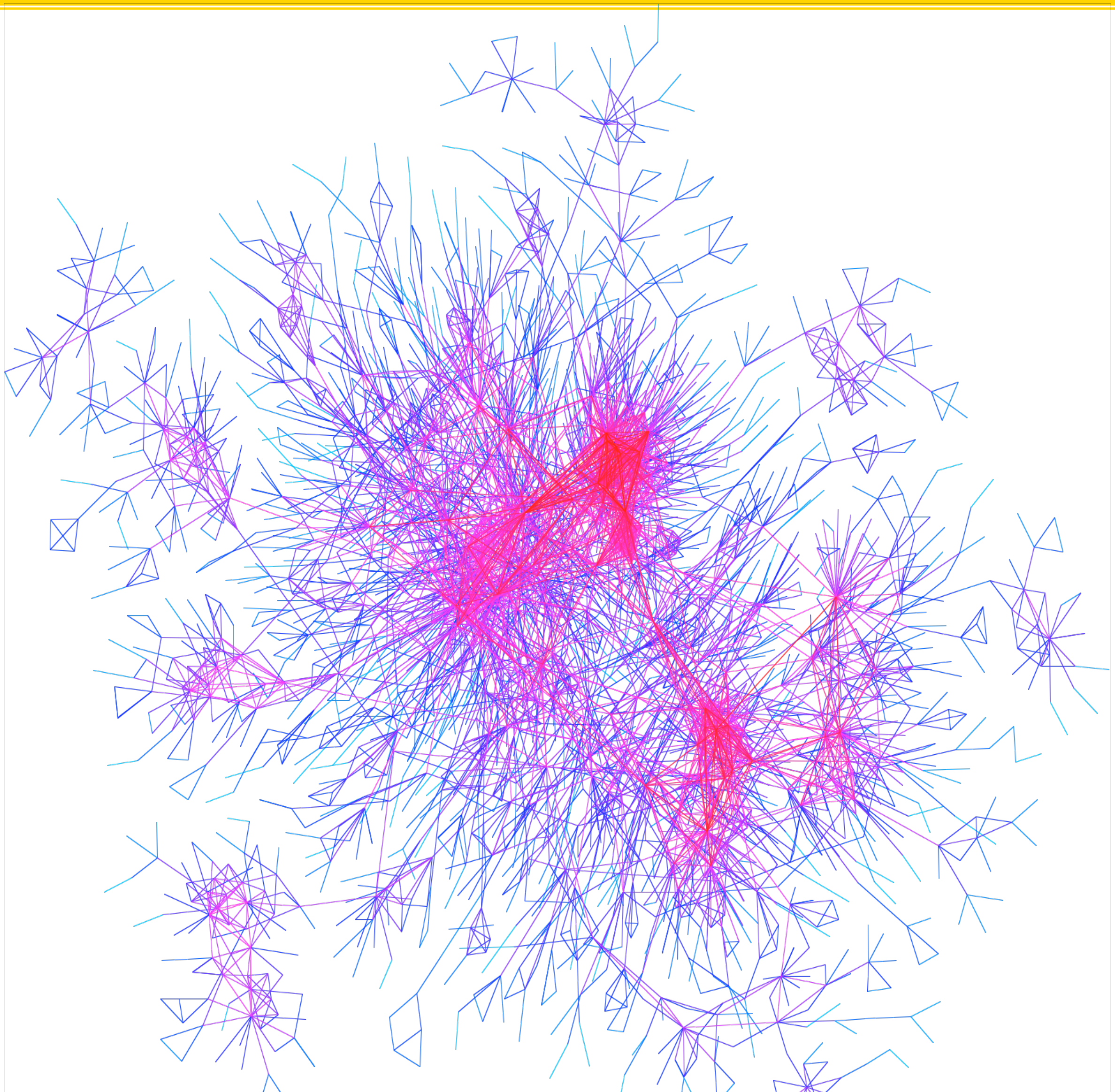
We prove that a good partition of the graph can be obtained by separating high ranked vertices from low ranked vertices.

The top 15500 phrases.

```
0        alameda flower
1000     flower in phoenix
2000     florist grange illinois la
3000     burnett florist
4000     flower macedonia
5000     florist flower
----------------------------
7000     cookie order
9000     day mother poem
11000    cream strawberry
*****************************
15500    margarita mix
```

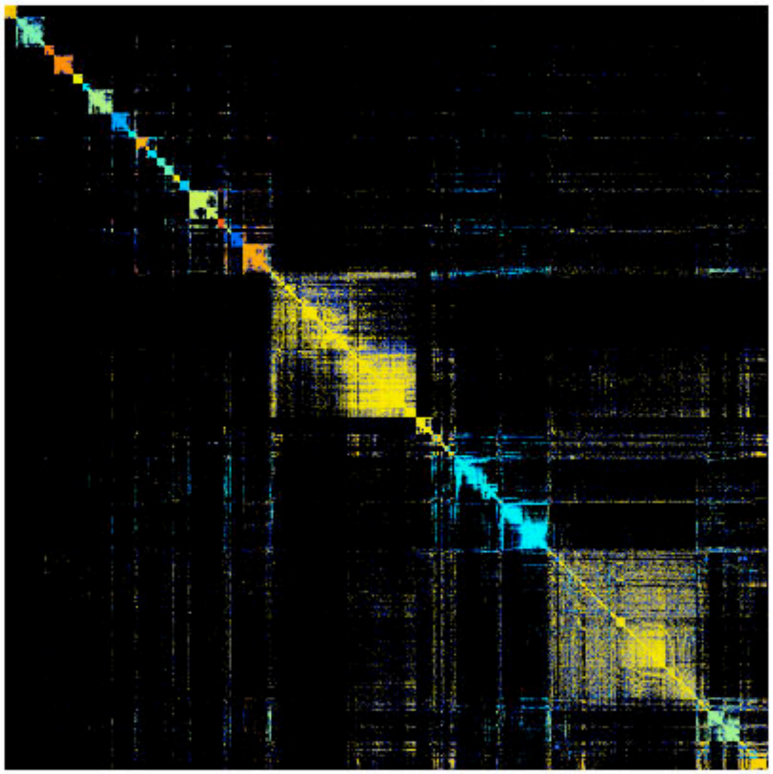

Results from queue_ppr.c on top30.eg2
top30.eig
cutsize
volume
30000
25000
20000
15000
10000
5000
0
0
200000
400000
600000
800000
1e+06
1.2e+06

Graph partitioning using PageRank vector.

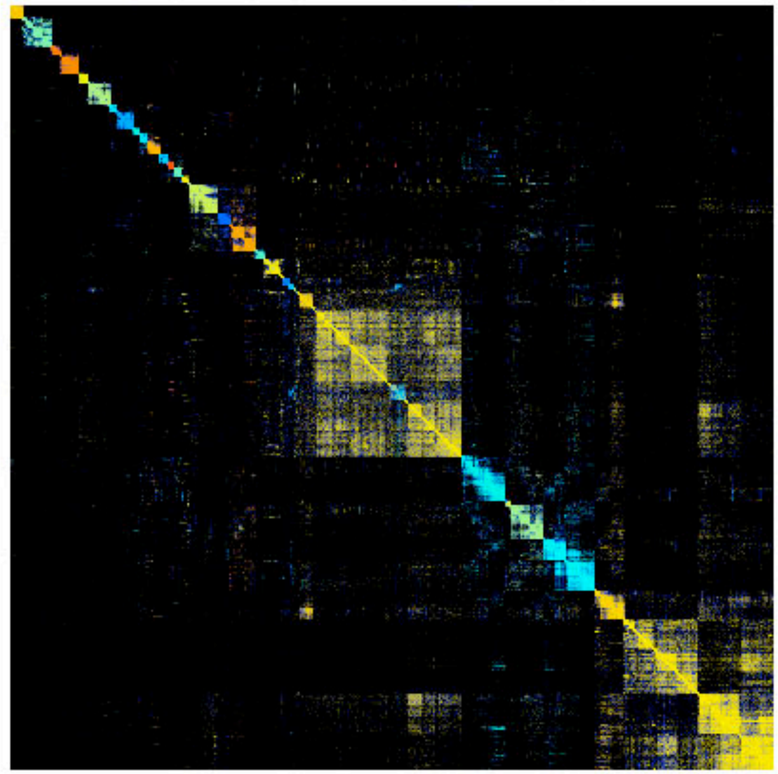


198,430 nodes and 1,133,512 edges

Internet Movie Database

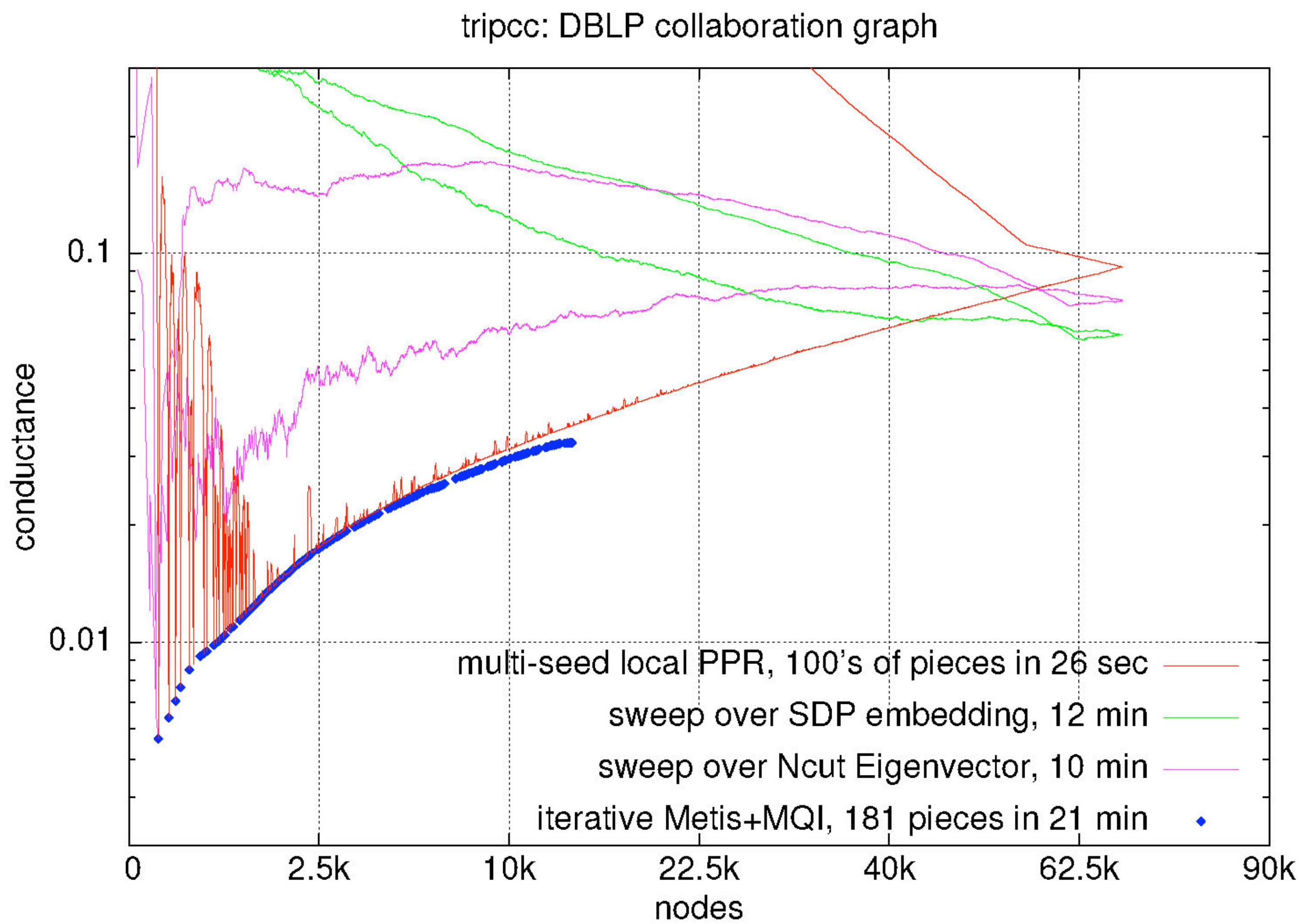


Local partitioning
(10 min)



Recursive spectral partitioning
(250 min)

Local PPR on DBLP graph



Kevin Lang 2007

4 Partitioning algorithm ⟷ 4 Cheeger inequalities

- graph spectral method — Fiedler '73, Cheeger, 60's
- random walks — Mihail 89; Lovasz, Simonovits, 90, 93; Spielman, Teng, 04
- PageRank — Andersen, Chung, Lang, 06
- heat kernel — Chung, PNAS , 08.

PageRank versus heat kernel

$$p_{\alpha,s} = \alpha \sum_{k=0}^{\infty} (1-\alpha)^k (sW^k)$$

Geometric sum

$$\rho_{t,s} = e^{-t} \sum_{k=0}^{\infty} s \frac{(tW)^k}{k!}$$

Exponential sum

PageRank versus heat kernel

$$p_{\alpha,s} = \alpha \sum_{k=0}^{\infty} (1-\alpha)^k (sW^k)$$

Geometric sum

$$\rho_{t,s} = e^{-t} \sum_{k=0}^{\infty} s \frac{(tW)^k}{k!}$$

Exponential sum

$$p = \alpha + (1-\alpha) pW$$

recurrence

$$\frac{\partial \rho}{\partial t} = -\rho (I - W)$$

Heat equation

Definition of heat kernel

$$H_t = e^{-t}(I + tW + \frac{t^2}{2}W^2 + \ldots + \frac{t^k}{k!}W^k + \ldots)$$

$$= e^{-t(I-W)}$$

$$= e^{-tL}$$

$$= I - tL + \frac{t^2}{2}L^2 + \ldots + (-1)^k \frac{t^k}{k!}L^k + \ldots$$

$$\frac{\partial}{\partial t}H_t = -(I-W)H_t$$

$$\rho_{t,s} = sH_t$$

Partitioning algorithm using the heat kernel

Theorem:

$$\left|\rho_{t,u}(S)-\pi(S)\right| \le \sqrt{\frac{vol(S)}{d_u}}e^{-t\kappa_{t,u}^2/4}$$

where $\kappa_{t,u}$ is the minimum Cheeger ratio over sweeps by using heat kernel pagerank over all u in S.

Partitioning algorithm using the heat kernel

Theorem:

$$\left|\rho_{t,u}(S)-\pi(S)\right| \le \sqrt{\frac{vol(S)}{d_u}}\,e^{-t\kappa_{t,u}^2/4}$$

where $\kappa_{t,u}$ is the minimum Cheeger ratio over sweeps by using heat kernel pagerank over all u in S.

Theorem: For $vol(S) \le vol(G)^{2/3}$,

$$\left|\rho_{t,S}(S)-\pi(S)\right| \ge e^{-th_S}.$$

(Improving the previous PageRank lower bound $1-t\,h_S$.)

Theorem:

$$\left|\rho_{t,S}(S)-\pi(S)\right| \geq (1-\pi(S))e^{-h_S t/(1-\pi(S))}$$

Sketch of a proof:

Consider $F(t) = -\log(\rho_{t,S}(S)-\pi(S))$

Show $\frac{\partial^2}{\partial t^2}F(t) \leq 0$

Then $\frac{\partial}{\partial t}F(t) \leq \frac{\partial}{\partial t}F(0) = \frac{\Phi_S}{1-\pi(S)}$

Solve and get $\left|\rho_{t,S}(S)-\pi(S)\right| \geq (1-\pi(S))e^{-h_S t/(1-\pi(S))}$

Partitioning algorithm using the heat kernel

Using the upper and lower bounds,

a Cheeger inequality can be obtained :

$$\Phi_S \geq \lambda_S \geq \frac{\kappa_S{}^2}{8} \geq \frac{\Phi_S{}^2}{8}$$

where λ_S is the Dirichlet eigenvalue of the Laplacian, and κ_S is the minimum Cheeger ratio over sweeps by using heat kernel with seeds S for appropriate t.

Random walks versus heat kernel

How fast is the convergence to the stationary distribution?

For what k, can one have

$$f\,W^k \to \pi \quad ?$$

Choose t to satisfy the required property.

Partitioning algorithm using the heat kernel

Using the upper and lower bounds,

a Cheeger inequality can be obtained :

$$\Phi_S \geq \lambda_S \geq \frac{\kappa_S{}^2}{8} \geq \frac{\Phi_S{}^2}{8}$$

where λ_S is the Dirichlet eigenvalue of the Laplacian, and κ_S is the minimum Cheeger ratio over sweeps by using heat kernel with seeds S for appropriate t.

Partitioning algorithm using the heat kernel

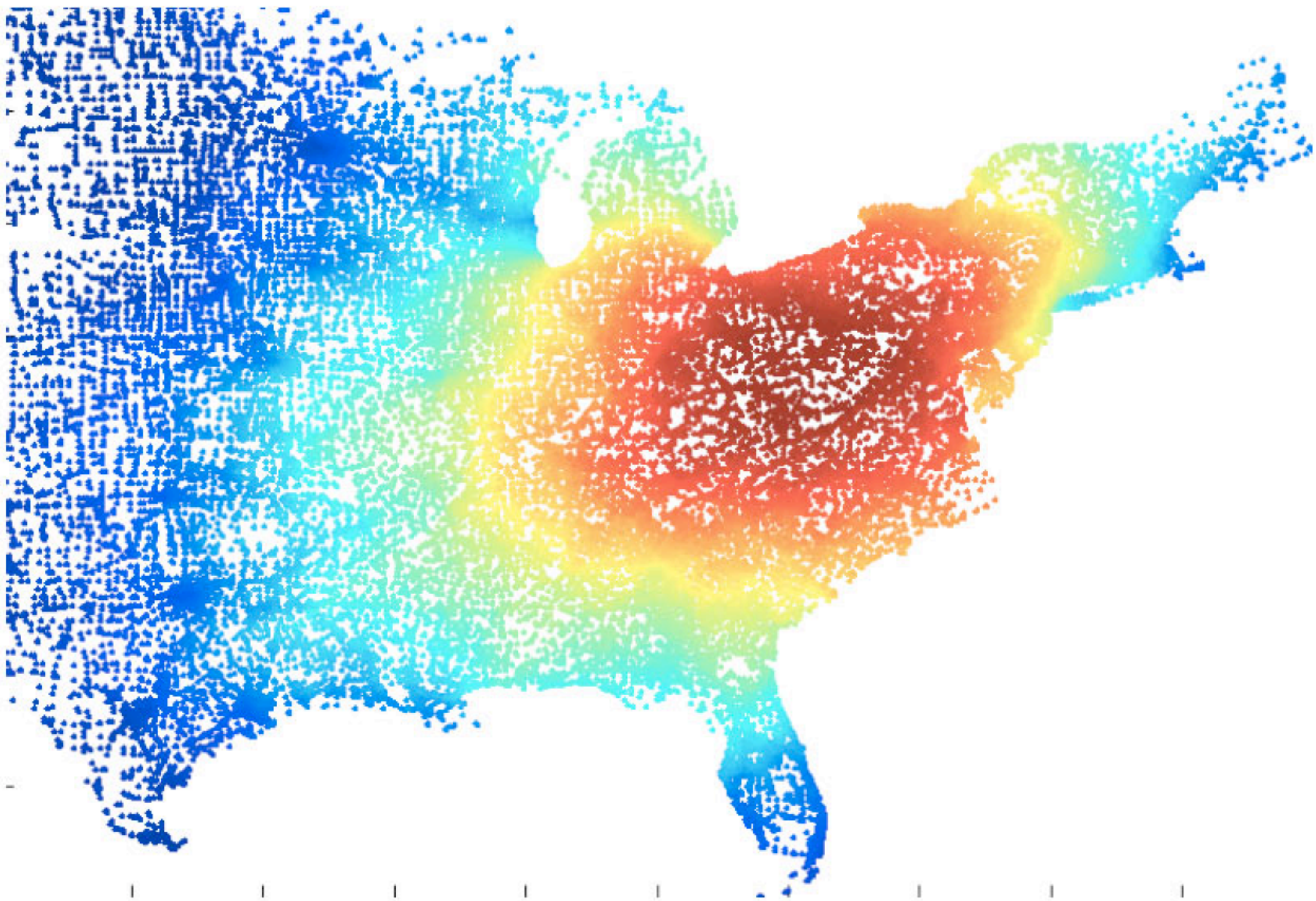
Using the upper and lower bounds,

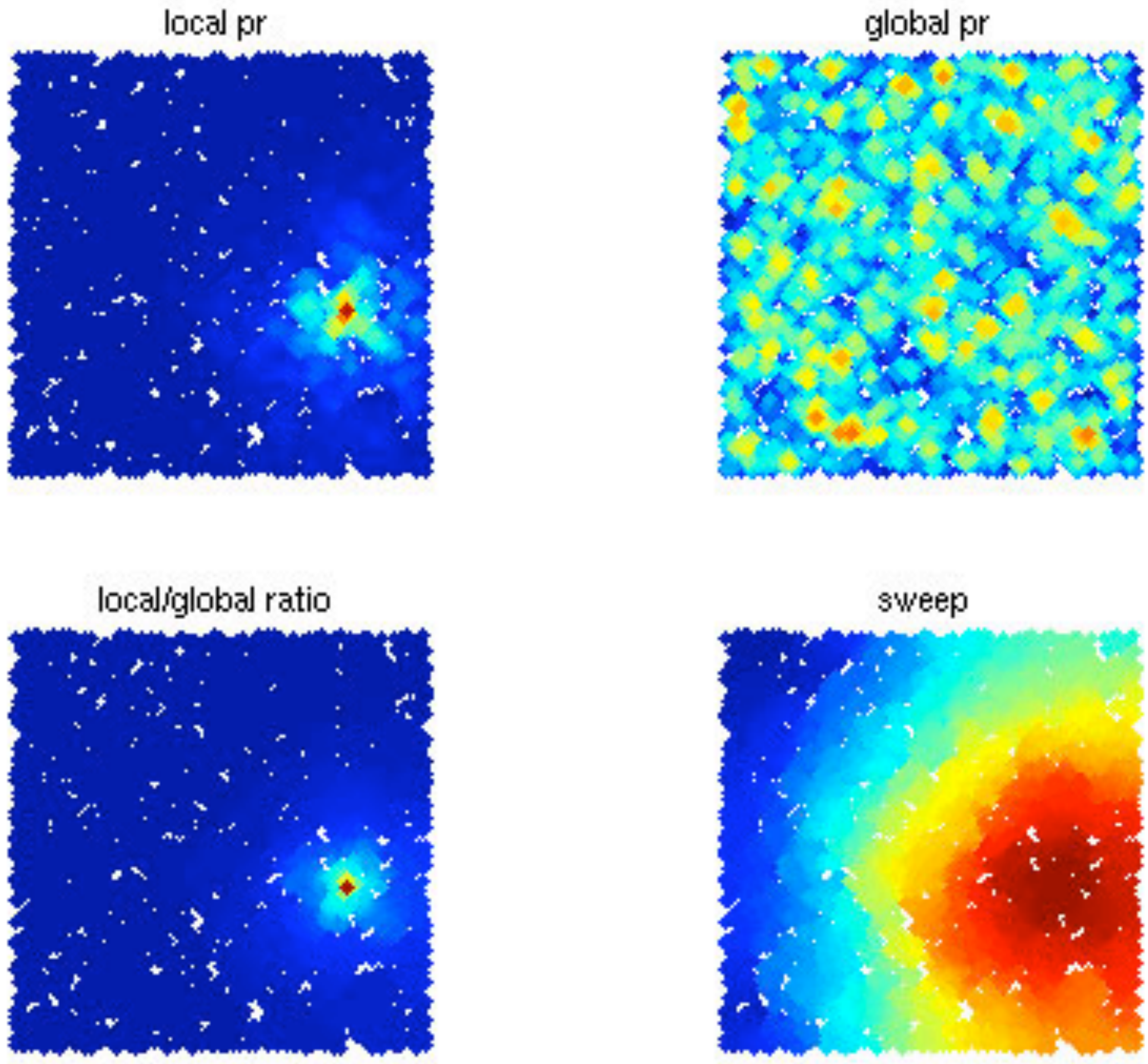
a Cheeger inequality can be obtained :

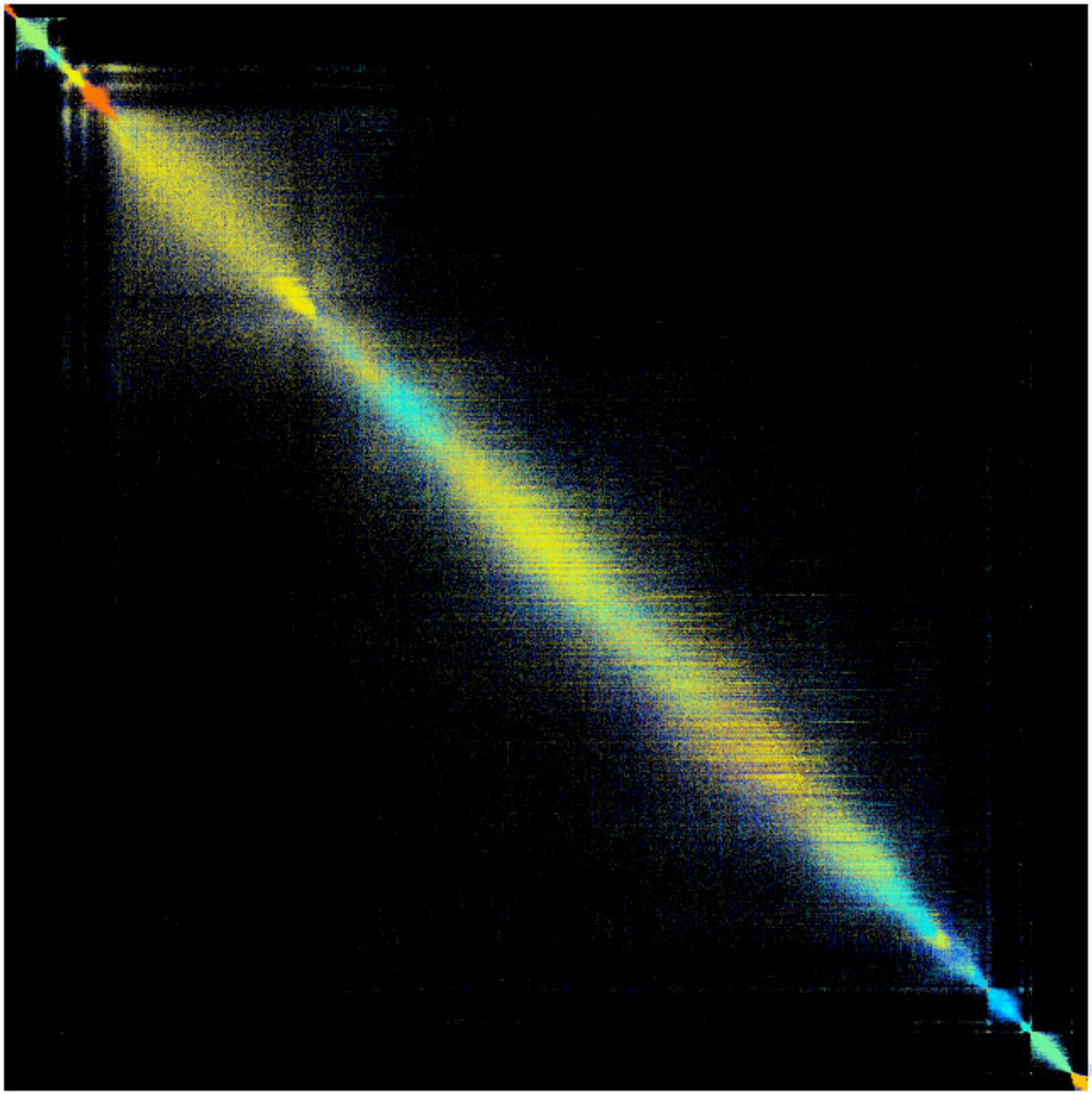
$$\Phi_S \geq \lambda_S \geq \frac{\kappa_u^{\,2}}{8\log s} \geq \frac{\Phi_u^{\,2}}{8\log s}$$

where λ_S is the Dirichlet eigenvalue of the Laplacian, and κ_u is the minimum Cheeger ratio over sweeps by using heat kernel with a random seed in S. The volume of the set of such u *is* > vol(S)/4.

What the sweep should look like

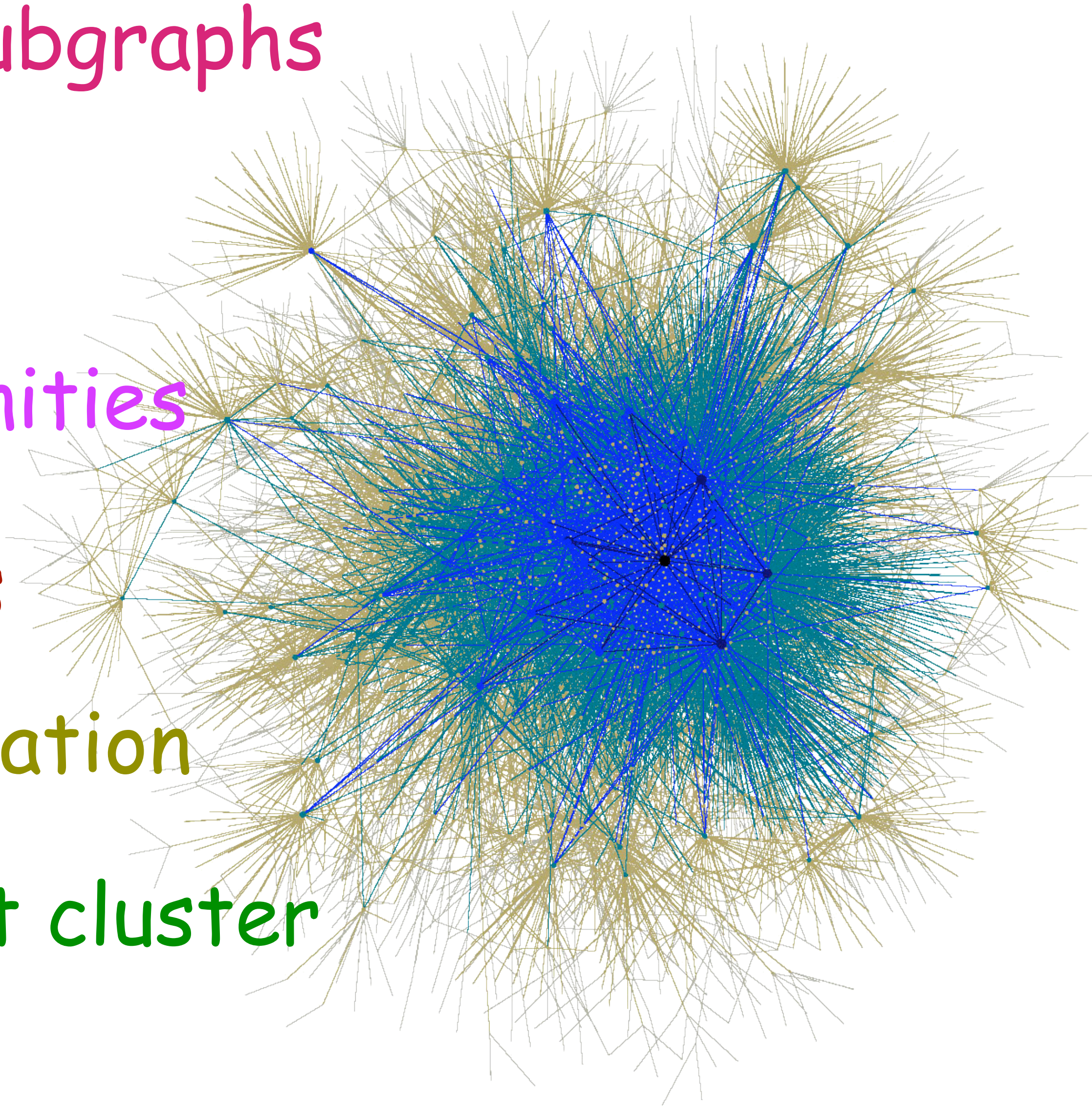



local pr
global pr
local/global ratio
sweep



Local partitioning algorithms

- Finding dense subgraphs
- Web search
- identify communities
- locate hot spots
- trace target location
- biological effect cluster
- …

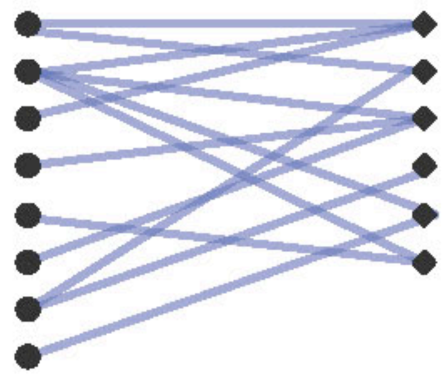


Some examples (Reid Andersen)

A bidding graph from Yahoo sponsored search

Phrases | Advertiser IDs

e.g. Margarita Mix | e.g. c8cbfd0bd74ba8cc



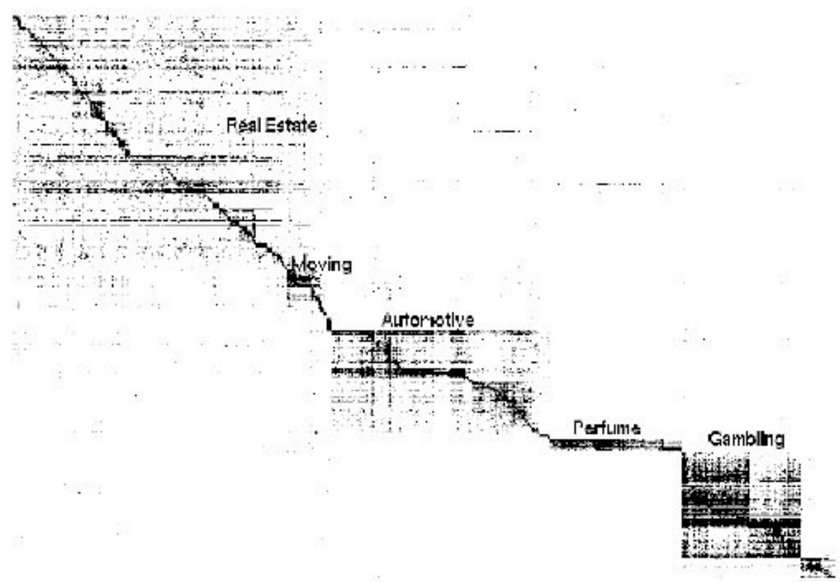
On the left are search phrases, on the right are advertisers.
Each edge represents a bid by an advertiser on a phrase.

$400K$ phrases, $200K$ advertisers, and 2 million edges.

Some examples (Reid Andersen)

Submarkets in the bigging graph

The bidding graph has numerous submarkets, related to real estate, flower delivery, hotels, gambling, ...



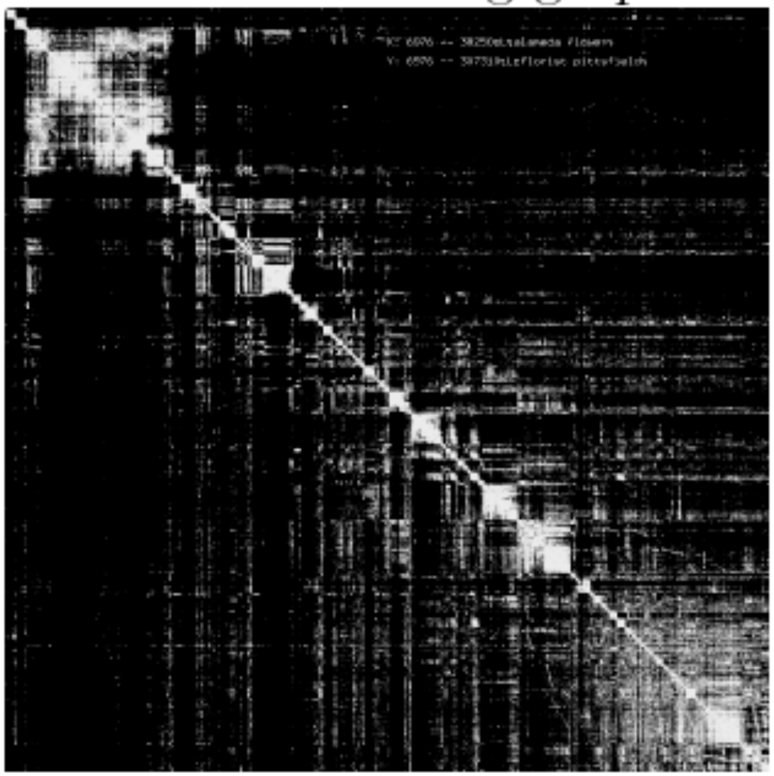
It is useful to identify these submarkets.

- Find groups of related phrases to suggest to advertisers.
- Find small submarkets for testing and experimentation.

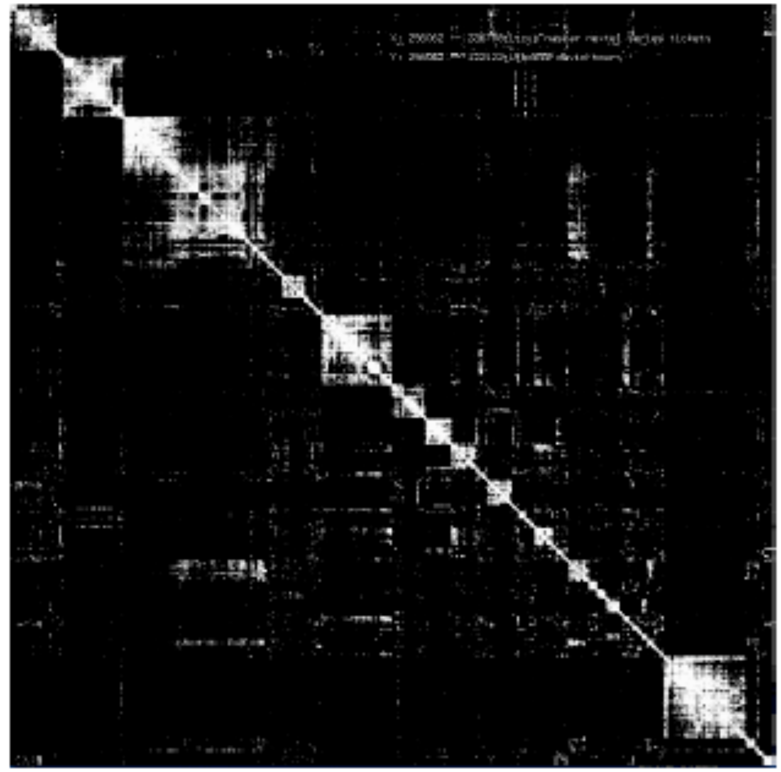
There are thousands of submarkets

If you want to decompose the graph into submarkets, the running time is determined by the time spent per vertex.

Bidding graph

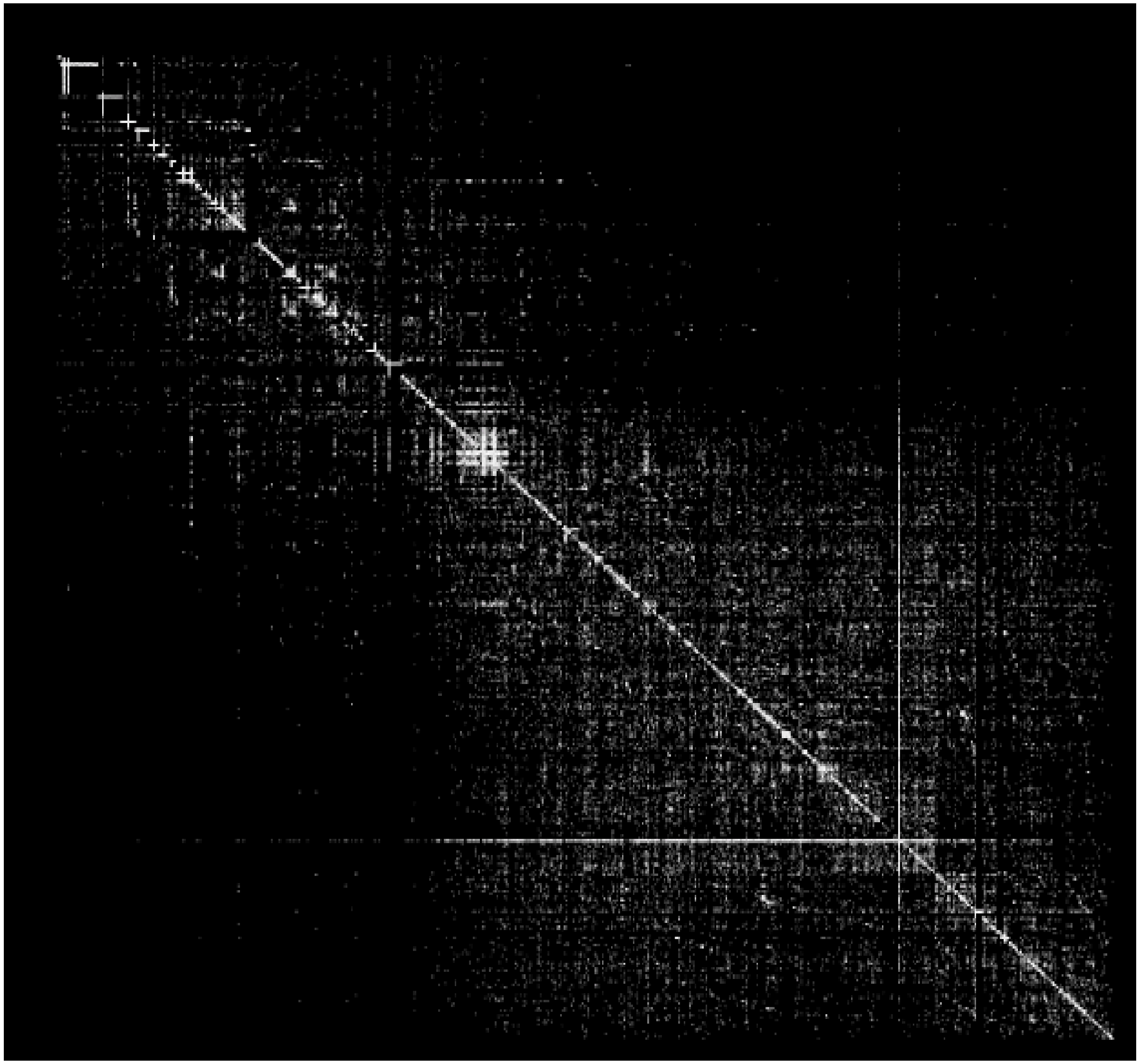


10x zoom



These pictures were made by applying local partitioning throughout the graph with random seed vertices and target sizes. The time spent per vertex is roughly $\log n/\Phi$.

A protein-protein interaction network



Recursive partitioning

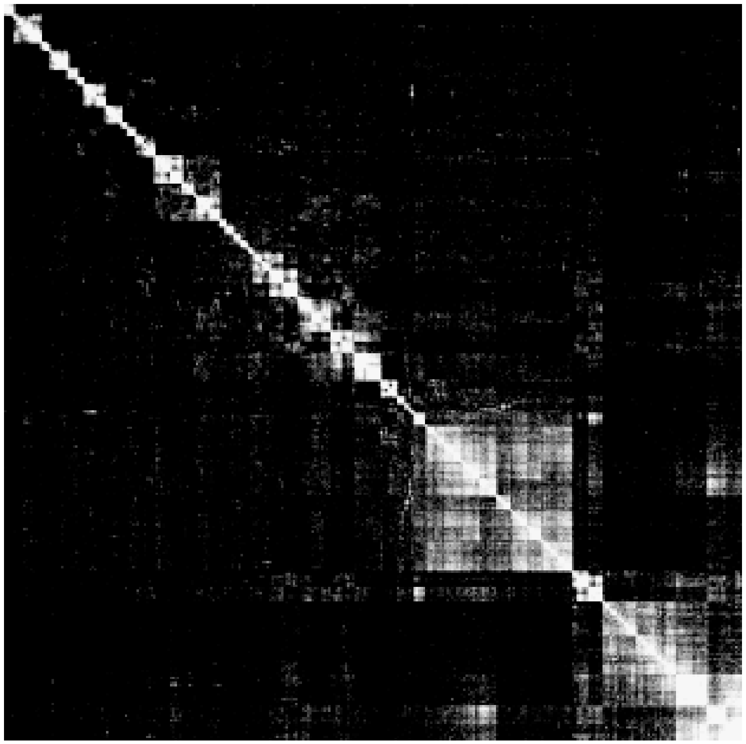


Figure: Internet Movie Database

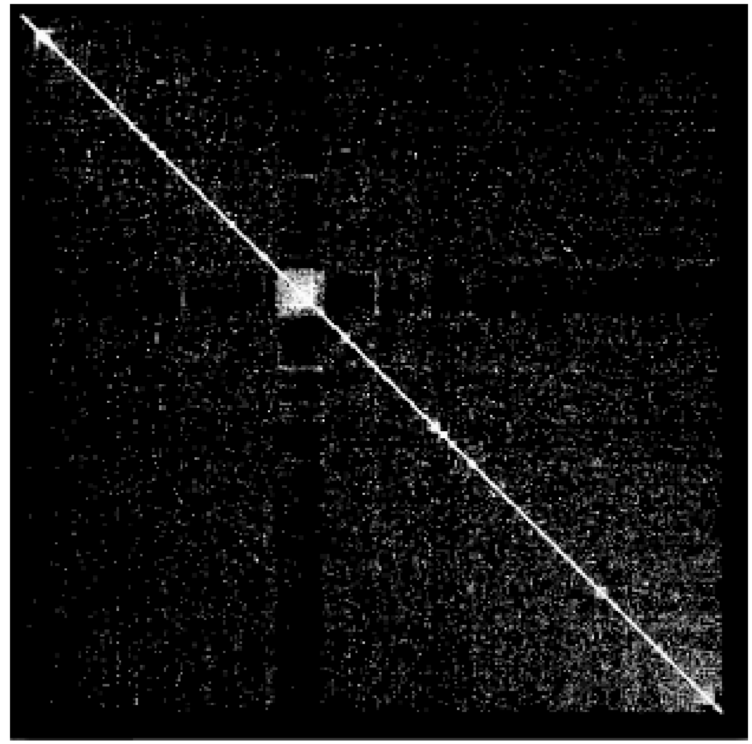


Figure: Instant Messenger subgraph

Algorithms based on our understanding for information networks

Topics:
- Complex networks
- pageranks
- Games on graphs

Methods:
- Probabilistic methods
- Analytic methods

Related areas:
- Spectral graph theory
- Random walks
- Random graphs
- Game theory
- Quasi-randomness

Some related papers:

- Andersen, Chung, Lang, Local graph partitioning using pagerank vectors, FOCS 2006
- Andersen, Chung, Lang, Local partitioning for directed graphs using PageRank, WAW 2007
- Chung, Four proofs of the Cheeger inequality and graph partition algorithms, ICCM 2007
- Chung, The heat kernel as the pagerank of a graph, PNAS 2008.