Four graph partitioning algorithms

Fan Chung

University of California, San Diego
History of graph partitioning

NP-hard $\rightarrow$ approximation algorithms

- Spectral method, Fiedler 73, Folklore
- Multicommmunity flow, Leighton+Rao 88
- Semidefinite programming, Arora+Rao+Vazirani 04
- Expander flow, Arora+Hazan+Kale 04
- Single commodity flows, Khandekar+Rao+Vazirani 06
“traditional” applications of graph partition algorithms:

Divide-and-conquer algorithms

- Circuit layout & designs
- Parallel computing
- Hierarchical clusterings
- Bioinformatics
- ...

Applications of partitioning algorithms for massive graphs

• Web search
• identify communities
• locate hot spots
• trace targets
• combat link spam
• epidemics
• ...

[Diagram of a complex network graph with nodes and edges representing connections and data relationships.]
1.5.6 Graph Partition
Excerpt from The Algorithm Design Manual: Graph partitioning arises as a preprocessing step to divide-and-conquer algorithms, where it is often a good idea ...
www.cs.sunysb.edu/~algorith/files/graph-partition.shtml - 19k - Cached - Similar pages

Algorithms and Software for Partitioning Graphs
Graph partitioning is an NP hard problem with numerous applications. ... An Improved Spectral Graph Partitioning Algorithm for Mapping Parallel Computations ...
www.sandia.gov/~bahendr/partitioning.html - 11k - Cached - Similar pages

Graph Partitioning
Then, the graph partitioning problem consists on dividing G into k disjoint partitions. The goal is minimize the number of cuts in the edges of the ...
www.ace.ual.es/~cgil/grafos/Graph_Partitioning.html - 12k - Cached - Similar pages

Graph partitioning - Wikipedia, the free encyclopedia
The graph partitioning problem in mathematics consists of dividing a graph into pieces, such that the pieces are of about the same size and there are few ...
en.wikipedia.org/wiki/Graph_partitioning - 16k - Cached - Similar pages
Outline of the talk

• Motivations

• Conductance and Cheeger’s inequality

• Four graph partitioning algorithms by using:
  - eigenvectors
  - random walks
  - PageRank
  - heat kernel

• Local graph algorithms

• Future directions
Two types of cuts:

- Vertex cut
- Edge cut

How “good” is the cut?
\[
\frac{e(S,V-S)}{\text{Vol } S} \quad \leftrightarrow \quad \frac{e(S,V-S)}{|S|}
\]

\[
\text{Vol } S = \sum_{v \in S} \deg(v) \quad \quad |S| = \sum_{v \in S} 1
\]
The Cheeger constant for graphs

The Cheeger constant

\[ \Phi_G = \min_s \frac{e(S, \overline{S})}{\min(\text{vol } S, \text{vol } \overline{S})} \]

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

\( \Phi_G \) and its variations are sometimes called “conductance”, “isoperimetric number”, ...
The Cheeger constant

\[ \Phi_G = \min_S \frac{e(S, \bar{S})}{\min(\text{vol } S, \text{vol } \bar{S})} \]

The Cheeger inequality

\[ 2\Phi_G \geq \lambda \geq \frac{\Phi_G^2}{2} \]

\( \lambda \) : 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
  \[ L = D - A \]
- Normalized Laplacian
  Random walks
  Rate of convergence
The spectrum of a graph

Discrete Laplace operator

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

\[ L(x, y) = \begin{cases} 
1 & \text{if } x = y \\
\frac{1}{d_x} & \text{if } x \neq y \text{ and } x \sim y
\end{cases} \]

not symmetric in general

• Normalized Laplacian

\[ L(x, y) = \begin{cases} 
1 & \text{if } x = y \\
\frac{1}{\sqrt{d_x d_y}} & \text{if } x \neq y \text{ and } x \sim y
\end{cases} \]

with eigenvalues

\[ 0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-1} \leq 2 \]
Can you hear the shape of a network?

\( \lambda \) dictates many properties of a graph.

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

How “good” is the cut by using the eigenvalue \( \lambda \)?
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.

\[ 2\Phi \geq \lambda \geq \frac{\Phi^2}{2} \]
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 $\lambda$ is the first non-trivial eigenvalue of the Laplacian and $\alpha$ is the minimum CHEEGER ratio in a sweep using the eigenvector $f$. 

Partitioning algorithm $\iff$ The CHEEGER inequality
Eigenvalue problem for $n \times n$ 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.)
Using a sweep by the eigenvector can reduce the exponential number of choices of subsets to a linear number.

Using a local sweep by random walks, PageRank and its variations can further reduce the a linear number of choices to a specified finite number of sizes.
Four one-sweep graph partitioning algorithms

- graph spectral method
- random walks
- PageRank
- heat kernel
4 Partitioning algorithm ↔ 4 Cheeger inequalities

- **graph spectral method**  Fiedler ’73, Cheeger, 60’s
- **random walks**  Lovasz, Simonovits, 90, 93
- **PageRank**  Spielman, Teng, 04
- **heat kernel**  Andersen, Chung, Lang, 06

Mihail 89
Chung, PNAS , 08.
Graph partitioning  ↔  Local graph partitioning

Courtesy of Reid Andersen
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.
The definition of PageRank given by Brin and Page is based on random walks.
History of computing Pagerank

• Brin+Page 98
• Personalized PageRank, Haveliwala 03
• Computing personalized PageRank,
  Jeh+Widom 03
  Berkhin 06
Random walks in a graph.

$G$: a graph

$P$: transition probability matrix

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

$d_u :=$ 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 $\alpha$

• or surf a linked webpage with probability $1 - \alpha$

$\alpha$ : the jumping constant

$$p = \alpha(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}) + (1 - \alpha)pW$$
Two equivalent ways to define PageRank $pr(\alpha,s)$

\begin{equation}
    p = \alpha s + (1 - \alpha) pW
\end{equation}

$S$: the seed as a row vector

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

\begin{align*}
(1) & \quad p = \alpha s + (1 - \alpha) pW \\
(2) & \quad p = \alpha \sum_{t=0}^{\infty} (1 - \alpha)^t (sW^t) \\

s & = \left( \frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n} \right) \quad \text{the (original) PageRank} \\

s & = \text{some “seed”, e.g.,} \quad (1, 0, \ldots, 0) \quad \text{personalized PageRank} \\

(\text{Organize the random walks by a scalar } \alpha.)
Partitioning algorithm using random walks

Mihail 89, Lovász+Simonovits, 90, 93

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

Leads to a Cheeger inequality:

\[ 2\Phi \geq \lambda \geq \frac{\beta_G^2}{8 \log n} \geq \frac{\Phi^2}{8 \log n} \]

where \( \beta_G \) is the minimum Cheeger ratio over sweeps by using a lazy walk of \( k \) steps from every vertex for an appropriate range of \( k \).
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
Partitioning algorithm using PageRank

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

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

where $\gamma_u$ is the minimum Cheeeger 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 \( \alpha=0.1/(\phi^2 b) \), \( \epsilon=2^{-b/b} \).

• One sweep algorithm using \( p \) for finding cuts with conductance < \( \phi \).

Performance analysis:

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

(Improving previous bounds by a factor of \( \phi\log s \).)
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
Internet Movie Database

Local partitioning
(10 min)

Recursive spectral partitioning
(250 min)

Courtesy of Reid Andersen
Local PPR on DBLP graph

tripcc: DBLP collaboration graph

multi-seed local PPR, 100’s of pieces in 26 sec
sweep over SDP embedding, 12 min
sweep over Ncut Eigenvector, 10 min
iterative Metis+MQI, 181 pieces in 21 min

Kevin Lang 2007
4 Partitioning algorithm $\leftrightarrow$ 4 Cheeger inequalities

- **graph spectral method** Fiedler ’73, Cheeger, 60’s
  Mihail 89

- **random walks** 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

\[ p = \alpha + (1-\alpha) pW \]

recurrence

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

Exponential sum

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

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:**

\[ |\rho_{t,u}(S) - \pi(S)| \leq \sqrt{\frac{\text{vol}(S)}{d_u}} e^{-t\kappa_{t,u}^2 / 4} \]

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:**

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

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

**Theorem:** For \( \text{vol}(S) \leq \text{vol}(G)^{2/3} \),

\[ \left| \rho_{t,S}(S) - \pi(S) \right| \geq e^{-th_S}. \]

(Improving the previous PageRank lower bound \( 1-th_S \).)
Theorem:

\[ |\rho_{t,S}(S) - \pi(S)| \geq (1 - \pi(S))e^{-h_s t / (1 - \pi(S))} \]

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 \[ |\rho_{t,S}(S) - \pi(S)| \geq (1 - \pi(S))e^{-h_s t / (1 - \pi(S))} \]
Random walks versus heat kernel

How fast is the convergence to the stationary distribution?

For what $k$, can one have $fW^k \rightarrow \pi$?

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 $\lambda_S$ is the Dirichlet eigenvalue of the Laplacian, and $\kappa_S$ is the minimum Cheeger ratio over sweeps by using heat kernel with seeds $S$ for appropriate $t$. 
Dirichlet eigenvalues for a subset \( S \subseteq V \)

\[
\lambda_S = \inf_f \frac{\sum (f(u) - f(v))^2}{\sum_w f(w)^2 d_w}
\]

over all \( f \) satisfying the Dirichlet boundary condition:

\[
f(v) = 0 \quad \text{for all } v \notin S.
\]
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 \( \lambda_S \) is the Dirichlet eigenvalue of the Laplacian, and \( \kappa_S \) is the minimum Cheeger ratio over sweeps by using heat kernel with seeds \( S \) for appropriate \( t \).
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 \( \lambda_S \) is the Dirichlet eigenvalue of the Laplacian, and \( \kappa_S \) 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 \( \lambda_S \) is the Dirichlet eigenvalue of the Laplacian, and \( \kappa_u \) 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

Courtesy of Reid Andersen
Future directions:

Use PageRank and the heat kernel pagerank to shed light on:

• The geometry of graphs?

• Solving combinatorial problems, such as covering, packing, matching, etc.

• Graph drawing, visualization

• Metric embedding ...
What the sweep should look like

Courtesy of Reid Andersen