Large Graph Mining - Patterns, Explanations and Cascade Analysis

Christos Faloutsos
CMU
Thank you!

- Foster Provost
- Sinan Aral
- Arun Sundararajan
- Shirley Lau
- Sara Gorecki
Graphs - why should we care?

>$10B revenue

>0.5B users

Food Web
[Martinez ’91]

Internet Map
[lumeta.com]
Roadmap

• Introduction – Motivation

• Part#1: Patterns in graphs
  – Some (power) laws
  – The 'no good cuts' shock
  – A possible explanation: fractals

• [Part#2: Cascade analysis]

• Conclusions
Solution# S.1

- Power law in the degree distribution [SIGCOMM99]

internet domains

\[ \log(\text{rank}) = -0.82 \cdot \log(\text{degree}) \]

\text{att.com} \quad \text{ibm.com}
Solution# S.2: Eigen Exponent $E$

A power law in the eigenvalues of the adjacency matrix

$E = -0.48$

May 2001

$\mathbf{A} \mathbf{x} = \lambda \mathbf{x}$

$E$ is the exponent of the power law.
Triangle Law: #S.3
[Tsourakakis ICDM 2008]

X-axis: degree
Y-axis: mean # triangles

n friends -> \(~n^{1.6}\) triangles
# MORE Graph Patterns

<table>
<thead>
<tr>
<th>Unweighted</th>
<th>Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Static</strong></td>
<td></td>
</tr>
<tr>
<td><strong>L02.</strong> Triangle Power Law (TPL) [Tsourakakis ’08]</td>
<td></td>
</tr>
<tr>
<td><strong>L03.</strong> Eigenvalue Power Law (EPL) [Siganos et al. ’03]</td>
<td></td>
</tr>
<tr>
<td><strong>L04.</strong> Community structure [Flake et al. ’02, Girvan and Newman ’02]</td>
<td></td>
</tr>
<tr>
<td><strong>Dynamic</strong></td>
<td></td>
</tr>
<tr>
<td><strong>L05.</strong> Densification Power Law (DPL) [Leskovec et al. ’05]</td>
<td><strong>L11.</strong> Weight Power Law (WPL) [McGlohon et al. ’08]</td>
</tr>
<tr>
<td><strong>L06.</strong> Small and shrinking diameter [Albert and Barabási ’99, Leskovec et al. ’05]</td>
<td></td>
</tr>
<tr>
<td><strong>L07.</strong> Constant size 2nd and 3rd connected components [McGlohon et al. ’08]</td>
<td></td>
</tr>
<tr>
<td><strong>L08.</strong> Principal Eigenvalue Power Law ($\lambda_1$PL) [Akoglu et al. ’08]</td>
<td></td>
</tr>
<tr>
<td><strong>L09.</strong> Bursty/self-similar edge/weight additions [Gomez and Santonia ’98, Gribble et al. ’98, Crovella and</td>
<td></td>
</tr>
</tbody>
</table>

**RTG: A Recursive Realistic Graph Generator using Random Typing** Leman Akoglu and Christos Faloutsos. *PKDD’09.*
# MORE Graph Patterns

<table>
<thead>
<tr>
<th></th>
<th>Unweighted</th>
<th>Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2. Triangle Power Law (TPL) [Tsourakakis `08]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Eigenvalue Power Law (EPL) [Siganos et al. `03]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. Community structure [Flake et al. <code>02, Girvan and Newman </code>02]</td>
<td></td>
</tr>
<tr>
<td><strong>Dynamic</strong></td>
<td>5. Densification Power Law (DPL) [Leskovec et al. `05]</td>
<td>✔️. Weight Power Law (WPL) [McGlohon et al. `08]</td>
</tr>
<tr>
<td></td>
<td>6. Small and shrinking diameter [Albert and Barabási <code>99, Leskovec et al. </code>05]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7. Constant size 2nd and 3rd connected components [McGlohon et al. `08]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8. Principal Eigenvalue Power Law ($\lambda_1$PL) [Akoglu et al. `08]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9. Bursty/self-similar edge/weight additions [Gomez and Santonia <code>98, Gribble et al. </code>98, Crovella and</td>
<td></td>
</tr>
</tbody>
</table>
MORE Graph Patterns

<table>
<thead>
<tr>
<th>Unweighted</th>
<th>Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>L02. Triangle Power Law (TPL) [Tsourakakis ’08]</td>
<td></td>
</tr>
<tr>
<td>L03. Eigenvector Power Law (EPL) [Liao et al. ’08]</td>
<td></td>
</tr>
<tr>
<td>L04. Community structure [Back et al. ’02, Girvan and Newman ’02]</td>
<td></td>
</tr>
<tr>
<td>L05. Densification Power Law (DPL) [Leskovec et al. ’05]</td>
<td></td>
</tr>
<tr>
<td>L06. Small and shrinking diameter [Albert and Barabasi ’99, Leskovec et al. ’05]</td>
<td></td>
</tr>
<tr>
<td>L07. Constant size 2nd and 3rd connected components [McGlohon et al. ’08]</td>
<td></td>
</tr>
<tr>
<td>L08. Principal Eigenvector Power Law (PEPL) [Akoglu et al. ’08]</td>
<td></td>
</tr>
<tr>
<td>L10. Snapshot Power Law (SPL) [McGlohon et al. ’08]</td>
<td></td>
</tr>
<tr>
<td>L11. Weight Power Law (WPL) [McGlohon et al. ’08]</td>
<td></td>
</tr>
</tbody>
</table>


WIN workshop, NYU (c) 2013, C. Faloutsos
Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
  – Some (power) laws
  – The 'no good cuts' shock
  – A possible explanation: fractals
• Part#2: Cascade analysis
• Conclusions

www.cs.cmu.edu/~christos/TALKS/13-10-WIN/
Background: Graph cut problem

- Given a graph, and \( k \)
- Break it into \( k \) (disjoint) communities
Graph cut problem

• Given a graph, and $k$
• Break it into $k$ (disjoint) communities
• (assume: block diagonal = ‘cavemen’ graph)

$k = 2$
Many algo’s for graph partitioning

- METIS [Karypis, Kumar +]
- 2nd eigenvector of Laplacian
- Modularity-based [Girwan+Newman]
- Max flow [Flake+]

...
Strange behavior of min cuts

• Subtle details: next
  – Preliminaries: min-cut plots of ‘usual’ graphs


“Min-cut” plot

- Do min-cuts recursively.

\[ \text{Mincut size} = \sqrt{N} \]

\[ \log (\text{mincut-size} / \#\text{edges}) \]

\[ \log (\# \text{ edges}) \]
“Min-cut” plot

• Do min-cuts recursively.

log (# edges) vs. log (mincut-size / #edges)

N nodes
“Min-cut” plot

• Do min-cuts recursively.

N nodes

New min-cut

Better cut

log (mincut-size / #edges)

Slope = -0.5

log (# edges)
"Min-cut" plot

For a d-dimensional grid, the slope is $-1/d$

For a random graph (and clique), the slope is 0
Experiments

• Datasets:
  – Google Web Graph: 916,428 nodes and 5,105,039 edges
  – Lucent Router Graph: Undirected graph of network routers from www.isi.edu/scan/mercator/maps.html; 112,969 nodes and 181,639 edges
  – User ➔ Website Clickstream Graph: 222,704 nodes and 952,580 edges

“Min-cut” plot

- What does it look like for a real-world graph?

log (mincut-size / #edges) vs. log (# edges)
Experiments

- Used the METIS algorithm [Karypis, Kumar, 1995]

- Google Web graph
- Values along the y-axis are averaged
- “lip” for large # edges
- Slope of -0.4, corresponds to a 2.5-dimensional grid!
Experiments

- Used the METIS algorithm [Karypis, Kumar, 1995]

  - Google Web graph
  - Values along the y-axis are averaged
  - “lip” for large # edges
  - Slope of -0.4, corresponds to a 2.5-dimensional grid!

Better cut

log (mincut-size / #edges)

log (# edges)
Experiments

- Same results for other graphs too…

Lucent Router graph

Clickstream graph
Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
  – Some (power) laws
  – The 'no good cuts' shock
  – A possible explanation: fractals
• Part#2: Cascade analysis
• Conclusions
2 Questions, one answer

• Q1: why so many power laws
• Q2: why no ‘good cuts’?
2 Questions, one answer

- Q1: why so many power laws
- Q2: why no ‘good cuts’?
- A: Self-similarity = fractals = ‘RMAT’ ~ ‘Kronecker graphs’
20” intro to fractals

- Remove the middle triangle; repeat
- \( \Rightarrow \) Sierpinski triangle
- (Bonus question - dimensionality?)
  - \( >1 \) (inf. perimeter – \( (4/3)^\infty \))
  - \( <2 \) (zero area – \( (3/4)^\infty \))
20” intro to fractals

Self-similarity -> no char. scale
-> power laws, eg:
2x the radius,
3x the #neighbors nn(r)

\[ nn(r) = C r^{\log_{2} \log_{3}} \]
20” intro to fractals

Self-similarity -> **no char. scale**
-> power laws, eg:
2x the radius,
3x the #neighbors \( \text{nn}(r) \)
\[ \text{nn}(r) = C \ r^{\log 3/\log 2} \]
20” intro to fractals

Self-similarity -> no char. scale
-> power laws, eg:
2x the radius,
3x the #neighbors

\[ nn = C r^{\log_3/\log_2} \]

2x the radius,
4x neighbors

\[ nn = C r^{\log_4/\log_2} = C r^2 \]
20”’ intro to fractals

Self-similarity -> no char. scale
-> power laws, eg:
2x the radius,
3x the #neighbors
\[ nn = C r^{\log_3/\log_2} \]
=1.58

Fractal dim.

2x the radius,
4x neighbors
\[ nn = C r^{\log_4/\log_2} = C r^2 \]
20” intro to fractals

**Self-similarity** -> no char. scale

-> **power laws**, eg:

2x the radius,

3x the #neighbors

\[ \text{nn} = C \cdot r^{\log_3/\log_2} \]

2x the radius,

4x neighbors

\[ \text{nn} = C \cdot r^{\log_4/\log_2} = C \cdot r^2 \]

Fractal dim.
How does self-similarity help in graphs?

- A: RMAT/Kronecker generators
  - With self-similarity, we get all power-laws, automatically,
  - And small/shrinking diameter
  - And ‘no good cuts’
Graph gen.: Problem dfn

- Given a growing graph with count of nodes $N_1$, $N_2$, ...
- Generate a realistic sequence of graphs that will obey all the patterns
  - Static Patterns
    - S1 Power Law Degree Distribution
    - S2 Power Law eigenvalue and eigenvector distribution
    - Small Diameter
  - Dynamic Patterns
    - T2 Growth Power Law (2x nodes; 3x edges)
    - T1 Shrinking/Stabilizing Diameters
Kronecker Graphs

\( G_1 \)

Adjacency matrix
Kronecker Graphs

Intermediate stage

Adjacency matrix

\[
\begin{pmatrix}
1 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 1 \\
\end{pmatrix}
\]
Kronecker Graphs

Intermediate stage

\[
\begin{pmatrix}
1 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 1
\end{pmatrix}
\]

\(G_1\)

Adjacency matrix

\[
\begin{pmatrix}
G_1 & G_1 & 0 \\
G_1 & G_1 & G_1 \\
0 & G_1 & G_1
\end{pmatrix}
\]

\(G_2 = G_1 \otimes G_1\)

Adjacency matrix
Kronecker Graphs

- Continuing multiplying with $G_1$ we obtain $G_4$ and so on ...

$G_4$ adjacency matrix
Kronecker Graphs

- Continuing multiplying with $G_1$ we obtain $G_4$ and so on …

$G_4$ adjacency matrix
Kronecker Graphs

• Continuing multiplying with $G_1$ we obtain $G_4$ and so on …

$G_4$ adjacency matrix
Kronecker Graphs

- Continuing multiplying with $G_1$ we obtain $G_4$ and so on …

Holes within holes; Communities within communities

$G_4$ adjacency matrix
Problem Definition

• Given a growing graph with nodes $N_1, N_2, \ldots$
• Generate a realistic sequence of graphs that will obey all the patterns
  – Static Patterns
    ✓ Power Law Degree Distribution
    ✓ Power Law eigenvalue and eigenvector distribution
    ✓ Small Diameter
  – Dynamic Patterns
    ✓ Growth Power Law
    ✓ Shrinking/Stabilizing Diameters
• First generator for which we can prove all these properties
Impact: Graph500

- Based on RMAT (= 2x2 Kronecker)
- Standard for graph benchmarks
- [http://www.graph500.org/](http://www.graph500.org/)
- Competitions 2x year, with all major entities: LLNL, Argonne, ITC-U. Tokyo, Riken, ORNL, Sandia, PSC, …

*To iterate is human, to recurse is devine*
Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
  – …
  – Q1: Why so many power laws?
  – Q2: Why no ‘good cuts’?
• Part#2: Cascade analysis
• Conclusions

A: real graphs -> self similar -> power laws

www.cs.cmu.edu/~christos/TALKS/13-10-WIN/
Kronecker Product – a Graph

- Continuing multiplying with $G_1$ we obtain $G_4$ and so on …
Kronecker Product – a Graph

• Continuing multiplying with $G_1$ we obtain $G_4$ and so on …

Communities within communities within communities …

How many Communities?
3?
9?
27?

$G_4$ adjacency matrix
Kronecker Product – a Graph

• Continuing multiplying with $G_1$ we obtain $G_4$ and so on …

Communities within communities within communities …

$G_4$ adjacency matrix

How many Communities?
3?
9?
27?

A: one – but not a typical, block-like community…
Communities?

(Gaussian) Clusters?

Piece-wise flat parts?

# songs

age

WIN workshop, NYU  (c) 2013, C. Faloutsos
Wrong questions to ask!
Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
  – …
  – Q1: The 'no good cuts' shock
  – Q2: Why no ‘good cuts’?
• What next?
• Conclusions

www.cs.cmu.edu/~christos/TALKS/13-10-WIN/
Challenge #1: ‘Connectome’ – brain wiring

- Which neurons get activated by ‘tomato’
- How wiring evolves
- Modeling epilepsy

Tom Mitchell  George Karypis  N. Sidiropoulos  V. Papalexakis

`glass’  `tomato’  `bell’

![Graph showing brain activity and seizures](graph.png)
Challenge #2: Time evolving networks / tensors

- Periodicities? Burstiness?
- What is ‘typical’ behavior of a node, over time
- Heterogeneous graphs (= nodes w/ attributes)
Summary

• *many* patterns in real graphs
  – Power-laws everywhere
  – ‘no good cuts’

• Self-similarity (RMAT/Kronecker): good model
Thanks

Disclaimer: All opinions are mine; not necessarily reflecting the opinions of the funding agencies

Thanks to: NSF IIS-0705359, IIS-0534205, CTA-INARC; Yahoo (M45), LLNL, IBM, SPRINT, Google, INTEL, HP, iLab
Project info: PEGASUS

www.cs.cmu.edu/~pegasus

Results on large graphs: with Pegasus + hadoop + M45

Apache license

Code, papers, manual, video

Prof. U Kang

Prof. Polo Chau
TAKE HOME MESSAGE:

Cross-disciplinarity

www.cs.cmu.edu/~christos/TALKS/13-10-WIN/