Large Graph Mining: Patterns, Cascades, Fraud Detection, and Algorithms

Christos Faloutsos
CMU
Thank you!

• Prof. Chin-Wan Chung
Thank you!

• Prof. Chin-Wan Chung
Roadmap

- Introduction – Motivation
  - Why study (big) graphs?
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
- Part#3: Cascades and immunization
- Conclusions
Graphs - why should we care?

~1B nodes (web sites)
~6B edges (http links)
‘YahooWeb graph’
Graphs - why should we care?

YahooWeb:
(a) In-degree vs. Out-degree
(b) Degree vs. Triangles
(c) Degree vs. PageRank

~1B nodes (web sites)
~6B edges (http links)
‘YahooWeb graph’

Graphs - why should we care?

>$10B; ~1B users
Graphs - why should we care?

Internet Map
[lumeta.com]

Food Web
[Martinez '91]
Graphs - why should we care?

• Power-grid!
  – Nodes: (plants/consumers)
  – Edges: power lines
Graphs - why should we care?

- web-log (‘blog’) news propagation
- computer network security: email/IP traffic and anomaly detection
- Recommendation systems
- ....

- Many-to-many db relationship -> graph
Motivating problems

• P1: patterns? Fraud detection?

• P2: patterns in time-evolving graphs / tensors

• P3: cascades – whom to immunize?
Roadmap

• Introduction – Motivation
  – Why study (big) graphs?
• Part#1: Patterns in graphs
• Part#2: time-evolving graphs; tensors
• Part#3: Cascades and immunization
• Conclusions
Part 1: Patterns, & fraud detection
Laws and patterns

• Q1: Are real graphs random?
Laws and patterns

• Q1: Are real graphs random?
• A1: NO!!
  – Diameter (‘6 degrees’; ‘Kevin Bacon’)
  – in- and out- degree distributions
  – other (surprising) patterns

• So, let’s look at the data
Solution# S.1

- Power law in the degree distribution [SIGCOMM99]

internet domains

\[
\log(\text{rank}) \quad \log(\text{degree})
\]

\[
\text{att.com} \quad \text{ibm.com}
\]
Solution# S.1

- Power law in the degree distribution
  [SIGCOMM99]

internet domains

log(degree)

log(rank)

-0.82
Solution# S.1

• Q: So what?

internet domains

log(degree)

att.com

-0.82

log(rank)

ibm.com
Solution# S.1

- Q: So what?

internet domains

log(degree) log(rank)

att.com

ibm.com

\(-0.82\)
Solution# S.1

• Q: So what?

= friends of friends (F.O.F.)

• A1: # of two-step-away pairs: 100^2 * N = 10 Trillion

internet domains

\[
\log(\text{degree}) \quad \log(\text{rank})
\]

-0.82
Solution# S.1

- Q: So what?
- A1: # of two-step-away pairs: \(100^2 \times N = 10\ \text{Trillion} \)

internet domains

\( \log(\text{rank}) \) vs \( \log(\text{degree}) \)

- att.com
- ibm.com

\(-0.82\)
Solution# S.1

- **Q:** So what?
- **A1:** \# of two-step-away pairs: \(O(d_{\text{max}}^2) \sim 10M^2\)

internet domains

\[\log(\text{rank})\]

\[\log(\text{degree})\]

\(-0.82\)

\~0.8PB -> a data center(!)

DCO @ CMU

WWW, Seoul (c) 2014, C. Faloutsos
Q: So what?

A1: # of two-step-away pairs: $O(d_{max}^2) \sim 10M^2$

-0.82

internet domains
att.com
ibm.com

~0.8PB -> a data center(!)

Gaussian trap

New algorithms}

Solution# S.1
Observation – big-data:

• $O(N^2)$ algorithms are \textasciitilde intractable - $N=1B$

• $N^2$ seconds $= 31B$ years (>2x age of universe)
Solution# S.2: Eigen Exponent $E$

A2: power law in the eigenvalues of the adjacency matrix (‘eig()’)

Exponent = slope
$E = -0.48$

May 2001

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

Rank of decreasing eigenvalue

Eigenvalue

Exponent = slope
Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
  – Patterns: Degree; Triangles
  – Anomaly/fraud detection
  – Graph understanding
• Part#2: time-evolving graphs; tensors
• Part#3: Cascades and immunization
• Conclusions
Solution# S.3: Triangle ‘Laws’

• Real social networks have a lot of triangles
Solution# S.3: Triangle ‘Laws’

• Real social networks have a lot of triangles
  – Friends of friends are friends
• Any patterns?
  – 2x the friends, 2x the triangles?
Triangle Law: #S.3
[Tsourakakis ICDM 2008]

Epinions, X-axis: degree
Y-axis: mean # triangles
$n$ friends $\rightarrow \sim n^{1.6}$ triangles

Reuters, X-axis: degree
Y-axis: mean # triangles
Slope 1.68

SN, X-axis: degree
Y-axis: mean # triangles
Slope 1.74
Triangle Law: Computations
[Tsourakakis ICDM 2008]

But: triangles are expensive to compute
(3-way join; several approx. algos) – $O(d_{\text{max}}^2)$

Q: Can we do that quickly?
A:
Triangle Law: Computations
[Tsourakakis ICDM 2008]

But: triangles are expensive to compute
(3-way join; several approx. algos) – $O(d_{\max}^2)$

Q: Can we do that quickly?
A: Yes!

$\#\text{triangles} = \frac{1}{6} \text{Sum} \left( \lambda_i^3 \right)$
(and, because of skewness (S2),
we only need the top few eigenvalues! - $O(E)$)
Triangle counting for large graphs?

Anomalous nodes in Twitter (~3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD’11]
Triangle counting for large graphs?

Anomalous nodes in Twitter(\(~ 3\) billion edges)

[U Kang, Brendan Meeder, +, PAKDD’11]
Triangle counting for large graphs?

Anomalous nodes in Twitter (~ 3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD’11]
Triangle counting for large graphs?

Anomalous nodes in Twitter (~ 3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD ’11]
# MORE Graph Patterns

<table>
<thead>
<tr>
<th>Static</th>
<th>Unweighted</th>
<th>Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️ 2. Triangle Power Law (TPL) [Tsourakakis ´08]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✔️ 3. Eigenvalue Power Law (EPL) [Siganos et al. ´03]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✔️ 4. Community structure [Flake et al. ´02, Girvan and Newman ´02]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic</td>
<td>L05. Densification Power Law (DPL) [Leskovec et al. ´05]</td>
<td>L11. Weight Power Law (WPL) [McGlohon et al. ´08]</td>
</tr>
<tr>
<td></td>
<td>L06. Small and shrinking diameter [Albert and Barabási ´99, Leskovec et al. ´05]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L07. Constant size 2nd and 3rd connected components [McGlohon et al. ´08]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L08. Principal Eigenvalue Power Law ($\lambda_1$PL) [Akoglu et al. ´08]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L09. 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>Static</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unweighted</td>
<td>Weighted</td>
</tr>
<tr>
<td>102. Triangle Power Law (TPL) [Sousa et al. '10]</td>
<td>106. Small and shrinking diameter (Albert and Barabasi '99, Leskovec et al. '05)</td>
</tr>
<tr>
<td>103. Eigenvector Power Law (EPL) [Signos et al. '03]</td>
<td>107. Constant size 2nd and 3rd connected components [McGlohon et al. '08]</td>
</tr>
<tr>
<td>104. Community structure [Flake et al. '02, Girvan and Newman '02]</td>
<td>108. Principal Eigenvector Power Law ($\lambda_1 P_L$) [Akoglu et al. '08]</td>
</tr>
<tr>
<td>109. Dense/regular subgraphs with small/regular edges [Gomez and Santoro '98, Giribet et al. '08, Crovella and Bestavros '99, McGlohon et al. '08]</td>
<td>111. Weight Power Law (WPL) [McGlohon et al. '08]</td>
</tr>
</tbody>
</table>


Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
  – Patterns
  – Anomaly / fraud detection
  – Graph understanding
• Part#2: time-evolving graphs; tensors
• Part#3: Cascades and immunization
• Conclusions
Fraud

• Given
  – Who ‘likes’ what page, and when

• Find
  – Suspicious users and suspicious products

Fraud

• Given
  – Who ‘likes’ what page, and when

• Find
  – Suspicious users and suspicious products
Ill-gotten Facebook Pages

- Popular Page = $

- Fake ‘likes’ through unethical means:
  - Fake accounts
  - Malware
  - Credential stealing
  - Social Engineering
Graph Patterns and Lockstep Behavior

Our intuition

- Lockstep behavior: Same Likes, same time
Graph Patterns and Lockstep Behavior

Our intuition

- **Lockstep behavior**: Same Likes, same time

![Diagram showing graph patterns and lockstep behavior](image)
Graph Patterns and Lockstep Behavior

Our intuition

- Lockstep behavior: Same Likes, same time

Suspicious Lockstep Behavior
MapReduce Overview

- Use Hadoop to search for many clusters in parallel:
  1. **Start** with randomly seed
  2. **Update** set of Pages and center Like times for each cluster
  3. **Repeat** until convergence
Deployment at Facebook

- *CopyCatch* runs regularly (along with many other security mechanisms, and a large Site Integrity team)

3 months of *CopyCatch* @ Facebook

#users caught
Deployment at Facebook

Manually labeled 22 randomly selected clusters from February 2013

Most clusters (77%) come from real but compromised users

Fake acct (23%)
Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
  – Patterns
  – Anomaly / fraud detection
  – Graph understanding
• Part#2: time-evolving graphs; tensors
• Part#3: Cascades and immunization
• Conclusions
• Nodes: editors
• Edge A->B: ‘A’ changed ‘B’
VoG: Summarizing and Understanding Large Graphs
Danai Koutra,
U Kang,
Jilles Vreeken,
Christos Faloutsos.

Code: www.cs.cmu.edu/~dkoutra/CODE/vog.tar
VoG: Summarizing Wiki-controversy

click here to zoom into the image.

Top-8 star structures: admins, heavy wiki users, bots

Warring factions changing each other's edits. (Kiev vs Kiyv)

Ditto, between vandals

WWW, Seoul

(c) 2014, C. Faloutsos
VoG: Summarizing Graphs using Rich Vocabularies

Main Ideas:

1. Use 'vocabulary' of subgraph types

   ![Subgraph Examples](image)

2. Minimum Description Length (MDL) and above vocabulary to summarize graph
Summary of Part #1

- *many* patterns in real graphs
  - Power-laws everywhere
  - Gaussian trap
    - Avg << Max
Roadmap

- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
  - P2.1: time-evolving graphs
  - P2.2: with side information (‘coupled’ M.T.F.)
  - Speed
- Part#3: Cascades and immunization
- Conclusions
Part 2:
Time evolving graphs; tensors
Graphs over time -> tensors!

• Problem #2.1:
  – Given who calls whom, and when
  – Find patterns / anomalies
Graphs over time -> tensors!

• Problem #2.1:
  – Given who calls whom, and when
  – Find patterns / anomalies
Graphs over time -> tensors!

• Problem #2.1:
  – Given who calls whom, and when
  – Find patterns / anomalies
Graphs over time -> tensors!

• Problem #2.1:
  – Given who calls whom, and when
  – Find patterns / anomalies
Graphs over time -> tensors!

- Problem #2.1’:
  - Given author-keyword-date
  - Find patterns / anomalies

MANY more settings, with >2 ‘modes’
Graphs over time -> tensors!

• Problem #2.1’’:
  – Given subject – verb – object facts
  – Find patterns / anomalies

MANY more settings, with >2 ‘modes’
Graphs over time -> tensors!

• Problem #2.1’’’:
  – Given <triplets>
  – Find patterns / anomalies

MANY more settings, with >2 ‘modes’ (and 4, 5, etc modes)
Graphs & side info

- Problem #2.2: coupled (e.g., side info)
  - Given subject – verb – object facts
  - And voxel-activity for each subject-word
  - Find patterns / anomalies

`apple tastes sweet`
Problem #2.2: coupled (eg., side info)
- Given subject – verb – object facts
- And voxel-activity for each subject-word
- Find patterns / anomalies
Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
• Part#2: time-evolving graphs; tensors
  – P2.1: time-evolving graphs
  – P2.2: with side information (‘coupled’ M.T.F.)
  – Speed
• Part#3: Cascades and immunization
• Conclusions
Answer to both: tensor factorization

• Recall: (SVD) matrix factorization: finds blocks

\( \mathbf{U}_1 \)  
\[ \sim \]  
\( \mathbf{U}_i \)
Answer to both: tensor factorization

- PARAFAC decomposition
Answer: tensor factorization

• PARAFAC decomposition
• Results for who-calls-whom-when
  – 4M x 15 days
Anomaly detection in time-evolving graphs

- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

~200 calls to EACH receiver on EACH day!
Anomaly detection in time-evolving graphs

- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

> ~200 calls to EACH receiver on EACH day!

WWW, Seoul (c) 2014, C. Faloutsos
Anomaly detection in time-evolving graphs

- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
• Part#2: time-evolving graphs; tensors
  – P2.1: Discoveries @ phonecall network
  – P2.2: Discoveries in neuro-semantics
  – Speed
• Part#3: Cascades and immunization
• Conclusions
Coupled Matrix-Tensor Factorization (CMTF)
Neuro-semantics

• **Brain Scan Data**
  - 9 persons
  - 60 nouns

• **Questions**
  - 218 questions
  - ‘is it alive?’, ‘can you eat it?’


Neuro-semantics

• **Brain Scan Data**
  • 9 persons
  • 60 nouns

• **Questions**
  • 218 questions
  • ‘is it alive?’, ‘can you eat it?’

Patterns?
**Neuro-semantics**

- **Brain Scan Data**
  - 9 persons
  - 60 nouns

- **Questions**
  - 218 questions
  - ‘is it alive?’, ‘can you eat it?’

**Patterns?**
Neuro-semantics
Neuro-semantics

Small items ->
Premotor cortex
Small items ->
Premotor cortex

Evangelos Papalexakis, Tom Mitchell, Nicholas Sidiropoulos, Christos Faloutsos, Partha Pratim Talukdar, Brian Murphy, Turbo-SMT: Accelerating Coupled Sparse Matrix-Tensor Factorizations by 200x, SDM 2014
Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
• Part#2: time-evolving graphs; tensors
  – P2.1: Discoveries @ phonecall network
  – P2.2: Discoveries in neuro-semantics
  – Speed
• Part#3: Cascades and immunization
• Conclusions
Speed of tensor/CMTF analysis

• Q1: Can we make it fast?
• Q2: Does it work for large, disk-based data?
A1: Turbo-SMT

Relative cost

200x faster

Relative run time

A1: Turbo-SMT

Exploiting sparsity

Relative cost

200x faster

Relative run time

Ideal

baseline
Q2: spilling to the disk?

Reminder: tensor (eg., Subject-verb-object)
144M non-zeros

26M subjects

48M verbs

26M objects

NELL (Never Ending Language Learner)
@CMU
A2: GigaTensor

Reminder: tensor (eg., Subject-verb-object)
26M x 48M x 26M, 144M non-zeros

NELL (Never Ending Language Learner)@CMU

U Kang, Evangelos E. Papalexakis, Abhay Harpale, Christos Faloutsos, GigaTensor: Scaling Tensor Analysis Up By 100 Times - Algorithms and Discoveries, KDD’12
A2: GigaTensor

- GigaTensor solves 100x larger problem

GigaTensor

Out of Memory

100x

Tensor Toolbox

Number of nonzero = I / 50
Part 2: Conclusions

• Time-evolving / heterogeneous graphs -> tensors
• PARAFAC finds patterns
• Turbo-SMT; GigaTensor -> fast & scalable
Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
• Part#2: time-evolving graphs; tensors
• Part#3: Cascades and immunization
• Conclusions
Part 3: Cascades & Immunization
Why do we care?

• Information Diffusion
• Viral Marketing
• Epidemiology and Public Health
• Cyber Security
• Human mobility
• Games and Virtual Worlds
• Ecology
• ..........
Roadmap

- A case for cross-disciplinarity
- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: Cascade analysis
  - (Fractional) Immunization
  - Epidemic thresholds
- Conclusions
Fractional Immunization of Networks
B. Aditya Prakash,
Lada Adamic,
Theodore Iwashyna (M.D.),
Hanghang Tong,
Christos Faloutsos
SDM 2013, Austin, TX
Whom to immunize?

- Dynamical Processes over networks

- Each circle is a hospital
- ~3,000 hospitals
- More than 30,000 patients transferred

Problem: Given $k$ units of disinfectant, whom to immunize?
Whom to immunize?

~6x fewer!

CURRENT PRACTICE

OUR METHOD

[US-MEDICARE NETWORK 2005]

Hospital-acquired inf. : 99K+ lives, $5B+ per year
Fractional Asymmetric Immunization

Drug-resistant Bacteria (like XDR-TB)

Hospital

Another Hospital

30%

(c) 2014, C. Faloutsos
Fractional Asymmetric Immunization
Fractional Asymmetric Immunization

Hospital → 15% → Another Hospital

WWW, Seoul (c) 2014, C. Faloutsos
Fractional Asymmetric Immunization

**Problem:**

Given $k$ units of disinfectant, distribute them to maximize hospitals saved.
Fractional Asymmetric Immunization

Problem:
Given $k$ units of disinfectant, distribute them to maximize hospitals saved @ 365 days.
Running Time

Wall-Clock Time

Simulations

SMART-ALLOC

> 1 week

> 30,000x speed-up!

14 secs

better

(c) 2014, C. Faloutsos
Experiments

# infected

K = 120

# epochs

uniform

better

SMART-ALLOC

WWW, Seoul

(c) 2014, C. Faloutsos
What is the ‘silver bullet’?

A: Try to decrease connectivity of graph

Q: how to measure connectivity?

– Avg degree? Max degree?
– Std degree / avg degree?
– Diameter?
– Modularity?
– ‘Conductance’ (~min cut size)?
– Some combination of above?
What is the ‘silver bullet’?

A: Try to decrease connectivity of graph

Q: how to measure connectivity?
A: first eigenvalue of adjacency matrix

Q1: why??
(Q2: dfn& intuition of eigenvalue ? )
Why eigenvalue?

A1: ‘G2’ theorem and ‘eigen-drop’:

• For (almost) any type of virus
• For any network
• -> no epidemic, if small-enough first eigenvalue \( (\lambda_1) \) of adjacency matrix
Why eigenvalue?

A1: ‘G2’ theorem and ‘eigen-drop’:

• For (almost) any type of virus
• For any network
• -> no epidemic, if small-enough first eigenvalue ($\lambda_1$) of adjacency matrix

• Heuristic: for immunization, try to min $\lambda_1$
• The smaller $\lambda_1$, the closer to extinction.
G2 theorem

Threshold Conditions for Arbitrary Cascade Models on Arbitrary Networks
B. Aditya Prakash, Deepayan Chakrabarti, Michalis Faloutsos, Nicholas Valler, Christos Faloutsos
IEEE ICDM 2011, Vancouver

extended version, in arxiv
http://arxiv.org/abs/1004.0060

~10 pages proof
Our thresholds for some models

- \( s = \text{effective strength} \)
- \( s < 1 : \text{below threshold} \)

<table>
<thead>
<tr>
<th>Models</th>
<th>Effective Strength (s)</th>
<th>Threshold (tipping point)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIS, SIR, SIRS, SEIR</td>
<td>( s = \lambda \left( \frac{\beta}{\delta} \right) )</td>
<td></td>
</tr>
<tr>
<td>SIV, SEIV</td>
<td>( s = \lambda \cdot \left( \frac{\beta \gamma}{\delta (\gamma + \theta)} \right) )</td>
<td>( s = 1 )</td>
</tr>
<tr>
<td>HIV</td>
<td>( s = \lambda \cdot \left( \frac{\beta_1 v_2 + \beta_2 \varepsilon}{v_2 (\varepsilon + v_1)} \right) )</td>
<td></td>
</tr>
</tbody>
</table>
Our thresholds for some models

- \( s = \text{effective strength} \)
- \( s < 1 : \text{below threshold} \)

### Models Effective Strength Threshold (tipping point)

<table>
<thead>
<tr>
<th>Model</th>
<th>Effective Strength</th>
<th>Threshold (tipping point)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIS, SIR, SIRS, SEIR</td>
<td>( s = \lambda )</td>
<td>( \left( \frac{\beta}{\delta} \right) )</td>
</tr>
<tr>
<td>SIV, SEIV w/ incubation</td>
<td>( s = \lambda )</td>
<td>( \left( \frac{\beta \gamma}{\delta(\gamma + \theta)} \right) )</td>
</tr>
<tr>
<td>H.I.V.</td>
<td>( s = \lambda )</td>
<td>( \left( \frac{\beta_1 v_2 + \beta_2 \varepsilon}{v_2(\varepsilon + v_1)} \right) )</td>
</tr>
</tbody>
</table>
Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
• Part#2: Cascade analysis
  – (Fractional) Immunization
  – intuition behind $\lambda_1$
• Conclusions
Intuition for $\lambda$

“Official” definitions:

- Let $A$ be the adjacency matrix. Then $\lambda$ is the root with the largest magnitude of the characteristic polynomial of $A$ [$\det(A - \lambda I)$].
- Also: $Ax = \lambda x$

Neither gives much intuition!

“Un-official” Intuition

- For ‘homogeneous’ graphs, $\lambda \approx$ degree

- $\lambda \sim$ avg degree
  - done right, for skewed degree distributions
Largest Eigenvalue ($\lambda$)

better connectivity $\rightarrow$ higher $\lambda$

$\lambda \approx 2$

(a) Chain

$\lambda = \sqrt{N}$

(b) Star

$\lambda = N-1$

(c) Clique

$N = 1000$ nodes

WWW, Seoul

(c) 2014, C. Faloutsos
Largest Eigenvalue ($\lambda$)

better connectivity $\rightarrow$ higher $\lambda$

(a) Chain

$\lambda \approx 2$

$\lambda = \sqrt{N}$

$N = 1000$ nodes

(b) Star

$\lambda = 31.67$

(c) Clique

$\lambda = 999$

WWW, Seoul

(c) 2014, C. Faloutsos
Examples: Simulations – SIR (mumps)

(a) Infection profile
PORTLAND graph: synthetic population, 31 million links, 6 million nodes

(b) “Take-off” plot
Examples: Simulations – SIRS (pertussis)

(a) Infection profile
PORTLAND graph: synthetic population, 31 million links, 6 million nodes

(b) “Take-off” plot
Part3: Immunization - conclusion

In \textit{(almost any)} immunization setting,

- Allocate resources, such that to
- \textbf{Minimize} $\lambda_1$
- \textit{(regardless of virus specifics)}

- Conversely, in a market penetration setting
  - Allocate resources to
  - \textbf{Maximize} $\lambda_1$
Roadmap

• Introduction – Motivation
  – Why study (big) graphs?
• Part#1: Patterns in graphs
• Part#2: time-evolving graphs; tensors
• Part#3: Cascades and immunization
• Future directions
• Conclusions
Brain connectivity

“apple” “Is it edible?” (y/n)
“knife” “Can it hurt you?” (y/n)

Frontal lobe (attention)

Parietal lobe (movement)

Occipital lobe (vision)

Temporal lobe (language)

MEG

voxel 1

voxel 2

voxel 306
Brain connectivity

“apple” “Is it edible?” (y/n)
“knife” “Can it hurt you?” (y/n)

Frontal lobe (attention)

Parietal lobe (movement)

Occipital lobe (vision)

Temporal lobe (language)

MEG

Fl  FL  TL  OL

voxel 1

voxel 2

voxel 306
Roadmap

• Introduction – Motivation
  – Why study (big) graphs?
• Part#1: Patterns in graphs
• Part#2: time-evolving graphs; tensors
• Part#3: Cascades and immunization
• Future directions
• Acknowledgements and Conclusions
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
CONCLUSION#1 – Big data

- Large datasets reveal patterns/outliers that are invisible otherwise
CONCLUSION#2 – tensors

• powerful tool
CONCLUSION#3 – eigen-drop

• Cascades & immunization: G2 theorem & eigenvalue

CURRENT PRACTICE

[US-MEDICARE NETWORK 2005]

OUR METHOD

~6x fewer!

> 30,000x speed-up!

14 secs
References

• D. Chakrabarti, C. Faloutsos: *Graph Mining – Laws, Tools and Case Studies*, Morgan Claypool 2012

TAKE HOME MESSAGE:

Cross-disciplinarity
Thank you! Questions?

Cross-disciplinarity

WWW, Seoul

(c) 2014, C. Faloutsos