Mining Large Graphs and Fraud Detection

Christos Faloutsos

CMU
Thank you!

Prof. Mohammad Hammoud

Nancy Lacson
Roadmap

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

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

~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?

- 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
Motivating problems

• P1: patterns? Fraud detection?

• P2: patterns in time-evolving graphs / tensors
Roadmap

• Introduction – Motivation
  – Why study (big) graphs?
• Part#1: Patterns & fraud detection
• Part#2: time-evolving graphs; tensors
• 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 [Faloutsos x 3 SIGCOMM99]

internet domains

log(degree) vs log(rank)

\[ \text{log(rank)} \leq 1.2 \times \text{log(degree)} \]

- Examples of internet domains:
  - att.com
  - ibm.com
Solution# S.1

• Power law in the degree distribution [Faloutsos x 3 SIGCOMM99]

internet domains

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

-0.82

att.com

ibm.com
Solution# S.2: Eigen Exponent $E$

- A2: power law in the eigenvalues of the adjacency matrix (`eig()`)

Exponent = slope

$E = -0.48$

May 2001
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]

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

n friends -> \( \sim n^{1.6} \) triangles
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!

#triangles = 1/6 \sum (\lambda_i^3)
(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]
Triangle counting for large graphs?

Anomalous nodes in Twitter (~ 3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD’11]
<table>
<thead>
<tr>
<th>Static</th>
<th>Unweighted</th>
<th>Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️ 22. Triangle Power Law (TPL) [Tsourakakis '08]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✔️ 23. Eigenvalue Power Law (EPL) [Siganos et al. '03]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L04. Community structure [Flake et al. '02, Girvan and Newman '02]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<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>L06. Small and shrinking diameter [Albert and Barabási '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 Eigenvalue Power Law ($\lambda_1$PL) [Akoglu et al. '08]</td>
<td></td>
</tr>
<tr>
<td>L09. Bursty/self-similar edge/weight additions [Gomez and Santonja '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>Unweighted</th>
<th>Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>200. Triangle Power Law (TPL) [Tsourakakis '08]</td>
<td></td>
</tr>
<tr>
<td>201. Eigenvalue Power Law (EPL) [Jajodia et al. '03]</td>
<td></td>
</tr>
<tr>
<td>202. Community structure [ Flake et al. '02, Girvan and Newman '02 ]</td>
<td></td>
</tr>
<tr>
<td>203. Denali Power Law (DPL) [Leskovec et al. '05]</td>
<td>204. Weight Power Law (WPL) [McGlohon et al. '08]</td>
</tr>
<tr>
<td>205. Small and shrinking diameter [Albert and Barabasi '99, Leskovec et al. '05]</td>
<td></td>
</tr>
<tr>
<td>206. Constant size 2nd and 3rd connected components [McGlohon et al. '08]</td>
<td></td>
</tr>
<tr>
<td>207. Principal Eigenvector Power Law ( Perez] [Akoglu et al. '08]</td>
<td></td>
</tr>
<tr>
<td>208. Bursty/self-similar edge/weight additions [Gomez and Santoro '06, Giribas et al. '08, Cowell and Estan '09, McGlohon et al. '08]</td>
<td></td>
</tr>
</tbody>
</table>

• Mary McGlohon, Leman Akoglu, Christos Faloutsos. *Statistical Properties of Social Networks*. in "Social Network Data Analytics" (Ed.: Charu Aggarwal)

Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
  – Patterns
  – Anomaly / fraud detection
    • CopyCatch
    • Spectral methods (‘fBox’)
    • Belief Propagation
• Part#2: time-evolving graphs; tensors
• 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

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
Graph Patterns and Lockstep Behavior

Our intuition

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

CMU-Q 2015 (c) 2015, C. Faloutsos
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

![Graph showing the number of users caught over time](image)
Deployment at Facebook

Fake acct

- Fake Accounts: 23%
- Malicious Browser Extensions: 9%
- OS Malware: 5%
- Credential Stealing: 5%
- Social Engineering: 5%

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

Manually labeled 22 randomly selected clusters from February 2013

CMU-Q 2015
(c) 2015, C. Faloutsos
Roadmap

• Introduction – Motivation

• Part#1: Patterns in graphs
  – Patterns
  – Anomaly / fraud detection
    • CopyCatch
    • Spectral methods (‘fBox’)
    • Belief Propagation

• Part#2: time-evolving graphs; tensors

• Conclusions
Problem: Social Network Link Fraud

Target: find “stealthy” attackers missed by other algorithms

Clique

Bipartite core

41.7M nodes
1.5B edges
Problem: Social Network Link Fraud

Target: find “stealthy” attackers missed by other algorithms

Takeaway: use reconstruction error between true/latent representation!

Roadmap

- Introduction – Motivation
- Part#1: Patterns in graphs
  - Patterns
  - Anomaly / fraud detection
    - CopyCatch
    - Spectral methods (‘fBox’)
    - Belief Propagation
- Part#2: time-evolving graphs; tensors
- Conclusions
E-bay Fraud detection

w/ Polo Chau & Shashank Pandit, CMU
[www’07]
E-bay Fraud detection
E-bay Fraud detection
E-bay Fraud detection - NetProbe

NetProbe alpha: Unearths Networks of Suspicious Auction Users

Inspects user 'alisher' for suspicious networks.

alisher

Registration: Aug. 13, 06
Location: United States

Fraudster: 95%
Accomplished: 4%
Hooded: 1%

Suspected fraudster -- this user has been behaving much like the other suspects by trading with the similar sets of possible accomplices.
Popular press

And less desirable attention:

• E-mail from ‘Belgium police’ (‘copy of your code?’)
Roadmap

• Introduction – Motivation

• Part#1: Patterns in graphs
  – Patterns
  – Anomaly / fraud detection
    • CopyCatch
    • Spectral methods (‘fBox’)
    • Belief Propagation; fast computation & unification

• Part#2: time-evolving graphs; tensors

• Conclusions
Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms

Danai Koutra
U Kang
Hsing-Kuo Kenneth Pao

Tai-You Ke
Duen Horng (Polo) Chau
Christos Faloutsos

ECML PKDD, 5-9 September 2011, Athens, Greece
Problem Definition:
GBA techniques

Given: Graph; & few labeled nodes
Find: labels of rest (assuming network effects)
Are they related?

• RWR (Random Walk with Restarts)
  – google’s pageRank (‘if my friends are important, I’m important, too’)

• SSL (Semi-supervised learning)
  – minimize the differences among neighbors

• BP (Belief propagation)
  – send messages to neighbors, on what you believe about them
Are they related? YES!

- **RWR (Random Walk with Restarts)**
  - google’s pageRank (‘if my friends are important, I’m important, too’)
- **SSL (Semi-supervised learning)**
  - minimize the differences among neighbors
- **BP (Belief propagation)**
  - send messages to neighbors, on what you believe about them
## Correspondence of Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Matrix</th>
<th>Unknown</th>
<th>known</th>
</tr>
</thead>
<tbody>
<tr>
<td>RWR</td>
<td>([\mathbf{I} - c \mathbf{A}\mathbf{D}^{-1}])</td>
<td>(\times) (x)</td>
<td>((1-c)y)</td>
</tr>
<tr>
<td>SSL</td>
<td>([\mathbf{I} + a(\mathbf{D} - \mathbf{A})])</td>
<td>(\times) (x)</td>
<td>(y)</td>
</tr>
<tr>
<td>FABP</td>
<td>([\mathbf{I} + a\mathbf{D} - c'\mathbf{A}])</td>
<td>(\times) (b_{h})</td>
<td>(\phi_{h})</td>
</tr>
</tbody>
</table>

- To each entry of the adjacency matrix \(\mathbf{A}\) in \(\phi_{h}\)
- *adjacency matrix*
- *final labels/beliefs*
- *prior labels/beliefs*
FABP is linear on the number of edges.
Results: Parallelism

FABP ~2x faster & wins/ties on accuracy.
Summary of Part #1

• *many* patterns in real graphs
  – Power-laws everywhere
  – Long (and growing) list of tools for anomaly/fraud detection
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 (eg., side info)
  – Given subject – verb – object facts
    • And voxel-activity for each subject-word
  – Find patterns / anomalies

`apple tastes sweet'
Graphs & side info

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

`apple tastes sweet'

`apple'

`tastes',

`sweet'

fMRI voxel activity

(c) 2015, C. Faloutsos
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
• Conclusions
Answer to both: tensor factorization

• Recall: (SVD) matrix factorization: finds blocks

\[ \sim \]

N users

M products

'Vegetarians' 'Plants'

'Kids' 'Cookies'

'Vegetarians' 'Plants'

'Kids' 'Cookies'

\[ \vec{u}_1 \]

\[ \vec{v}_1 \]

\[ \vec{u}_i \]
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
  - 1 caller
  - 5 receivers
  - 4 days of activity

~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!
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
• 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?**

- Airplane
- Dog
- Questions

CMU-Q 2015

(c) 2015, C. Faloutsos
**Nouns**
- Group 1: beetle, pants, bee
- Group 2: bear, cow, coat
- Group 3: glass, tomato, bell
- Group 4: bed, house, car

**Questions**
- Group 1:
  - can it cause you pain?
  - do you see it daily?
  - is it conscious?
- Group 2:
  - does it grow?
  - is it alive?
  - was it ever alive?
- Group 3:
  - can you pick it up?
  - can you hold it in one hand?
  - is it smaller than a golfball?
- Group 4:
  - does it use electricity?
  - can you sit on it?
  - does it cast a shadow?

**Brain Activity Maps**
- Group 1: Shows activation in the premotor cortex. Premotor Cortex

---

**Figure 4:**
[Image depicting brain activity maps and their corresponding questions.]
Neuro-semantics

Small items -> Premotor cortex

Nouns
- glass
- tomato
- bell

Questions
- can you pick it up?
- can you hold it in one hand?
- is it smaller than a golfball?’

Group 3

Premotor Cortex
Neuro-semantics

Small items -> Premotor cortex

Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
• Part#2: time-evolving graphs; tensors
  – P2.1: Discoveries at phonecall network
  – P2.2: Discoveries in neuro-semantics
  – Scalability & Speed
• Conclusions
Q: spilling to the disk?

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

48M verbs
26M subjects
26M objects

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

- GigaTensor solves $100x$ larger problem

Number of nonzero = $I / 50$
Part 2: Conclusions

- Time-evolving / heterogeneous graphs -> tensors
- PARAFAC finds patterns
- GigaTensor -> fast & scalable
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
CONCLUSION#1 – Big data

• Patterns \& Anomalies

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

• powerful tool

$$\text{Nouns}$$
- glass
- tomato
- bell

$$\text{Questions}$$
- can you pick it up?
- can you hold it in one hand?
- is it smaller than a golfball?'

Group 3

Premotor Cortex

CMU-Q 2015
(c) 2015, C. Faloutsos
References

• D. Chakrabarti, C. Faloutsos: *Graph Mining – Laws, Tools and Case Studies*, Morgan Claypool 2012
• http://www.morganclaypool.com/doi/abs/10.2200/S00449ED1V01Y201209DMK006
TAKE HOME MESSAGE:

Cross-disciplinarity

CMU-Q 2015  (c) 2015, C. Faloutsos
Thank you!

Cross-disciplinarity

CMU-Q 2015