Anomaly detection in large graphs

Christos Faloutsos
CMU
Thank you!

- Prof. Yu-Ru Lin
- Prof. Kostas Pelechrinis
‘Hi’ to db colleagues

• Prof. Vladimir Zadorozhny

• Prof. Panos Chrysanthis

• Prof. Alexandros Labrinidis
Roadmap

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

destination

source
time
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

* Robust Random Cut Forest Based Anomaly Detection on Streams* 
Sudipto Guha, Nina Mishra, Gourav Roy, Okke Schrijvers, ICML’16
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)

att.com

ibm.com
Solution# S.1

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

internet domains

\[
\log(\text{degree}) = a \times \log(\text{rank}) + b
\]

-0.82
### S2: connected component sizes

- **Connected Components** – 4 observations:

<table>
<thead>
<tr>
<th>Size</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4B nodes</td>
<td>6B edges</td>
</tr>
</tbody>
</table>

![Graph showing the distribution of connected component sizes](image)

*YahooWeb*

*Giant Connected Component*
S2: connected component sizes

• Connected Components

![Graph showing connected component sizes](attachment:image.png)

- 1) 10K x larger than next
S2: connected component sizes

- Connected Components

2) ~0.7B singleton nodes
S2: connected component sizes

- Connected Components

Count

3) SLOPE!
S2: connected component sizes

- Connected Components

4) Spikes!
S2: connected component sizes

- Connected Components

![Graph showing connected component sizes](image)

- YahooWeb

- suspicious financial-advice sites (not existing now)

- Giant Connected Component
Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
  – P1.1: Patterns: Degree; Triangles
  – P1.2: Anomaly/fraud detection
• Part#2: time-evolving graphs; tensors
• 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]

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

Reuters

Epinions

U.Pitt, 2/24/17

(c) C. Faloutsos, 2017
Triangle counting for large graphs?

Anomalous nodes in Twitter (~3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD’11]

U.Pitt, 2/24/17

(c) C. Faloutsos, 2017
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]

U.Pitt, 2/24/17

(c) C. Faloutsos, 2017
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]

U.Pitt, 2/24/17
(c) C. Faloutsos, 2017
S4: k-core patterns - dfn

- **k-core** (of a graph)
- **degeneracy** (of a graph)
- **coreness** (of a vertex)
CoreScope: Graph Mining Using k-Core Analysis - Patterns, Anomalies, and Algorithms

ICDM’16
Kijung Shin, Tina Eliassi-Rad and CF
Mirror Pattern: Observation

- **coreness** (of a vertex): maximum $k$ such that the vertex belongs to the $k$-core

- Definition: [Mirror Pattern] $\text{degree} \sim \text{coreness}$

Rank correlation $\rho = 0.99$
Mirror Pattern: Application

• Exceptions are ‘strange’
# MORE Graph Patterns

<table>
<thead>
<tr>
<th>Static</th>
<th>Unweighted</th>
<th>Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️.2</td>
<td>Triangle Power Law (TPL) [Tsourakakis `08]</td>
<td></td>
</tr>
<tr>
<td>✔️.3</td>
<td>Eigenvalue Power Law (EPL) [Siganos et al. `03]</td>
<td></td>
</tr>
<tr>
<td>✔️.4</td>
<td>Community structure [Flake et al. <code>02, Girvan and Newman </code>02]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic</th>
<th>Unweighted</th>
<th>Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️.5</td>
<td>Densification Power Law (DPL) [Leskovec et al. `05]</td>
<td>✔️.11. Weight Power Law (WPL) [McGlohon et al. `08]</td>
</tr>
<tr>
<td>✔️.6</td>
<td>Small and shrinking diameter [Albert and Barabási <code>99, Leskovec et al. </code>05]</td>
<td></td>
</tr>
<tr>
<td>✔️.7</td>
<td>Constant size 2\textsuperscript{nd} and 3\textsuperscript{rd} connected components [McGlohon et al. `08]</td>
<td></td>
</tr>
<tr>
<td>✔️.8</td>
<td>Principal Eigenvalue Power Law ($\lambda_1$PL) [Akoglu et al. `08]</td>
<td></td>
</tr>
<tr>
<td>✔️.9</td>
<td>Bursty/self-similar edge/weight additions [Gomez and Santonja <code>98, Gribble et al. </code>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>
</table>

Static

Dynamic

- 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
  - P1.1: Patterns
  - P1.2: Anomaly / fraud detection
    - No labels – spectral
    - With labels: Belief Propagation
- Part#2: time-evolving graphs; tensors
- Conclusions
How to find ‘suspicious’ groups?

• ‘blocks’ are normal, right?
Except that:

- ‘blocks’ are normal, right?
- ‘hyperbolic’ communities are more realistic
  [Araujo+, PKDD’14]
Except that:

- ‘blocks’ are usually **suspicious**
- ‘hyperbolic’ communities are more realistic

[**Araujo+**, PKDD’14]

Q: Can we spot blocks, easily?
Except that:

- ‘blocks’ are usually suspicious
- ‘hyperbolic’ communities are more realistic [Araujo+, PKDD’14]

Q: Can we spot blocks, easily?
A: Silver bullet: SVD!
Crush intro to SVD

- Recall: (SVD) matrix factorization: finds blocks

\[ \mathbf{U} \mathbf{P} \mathbf{V}^T \]

- ‘music lovers’ ‘sports lovers’ ‘citizens’
- ‘singers’ ‘athletes’ ‘politicians’

\[ \mathbf{U} \mathbf{V}^T \]

\[ \mathbf{U}_1 \mathbf{V}_1^T \]

\[ \mathbf{U}_i \mathbf{V}_i^T \]

U.Pitt, 2/24/17

(c) C. Faloutsos, 2017
Crush intro to SVD

- Recall: (SVD) matrix factorization: finds blocks

M fans

N idols

N 'music lovers' 'sports lovers' 'citizens'

M 'singers' 'athletes' 'politicians'

\[ U \approx U_1 U_1^T + U_i U_i^T \]

U.Pitt, 2/24/17

(c) C. Faloutsos, 2017
Inferring Strange Behavior from Connectivity Pattern in Social Networks

PAKDD’14

Meng Jiang, Peng Cui, Shiqiang Yang (Tsinghua)
Alex Beutel, Christos Faloutsos (CMU)
Lockstep and Spectral Subspace Plot

- Case #0: No lockstep behavior in random power law graph of 1M nodes, 3M edges
- Random ↔ “Scatter”
Lockstep and Spectral Subspace Plot

- Case #1: non-overlapping lockstep
- “Blocks” ←→ “Rays”

Rule 1 (short “rays”): two blocks, high density (90%), no “camouflage”, no “fame”

U.Pitt, 2/24/17
(c) C. Faloutsos, 2017
Lockstep and Spectral Subspace Plot

- Case #2: non-overlapping lockstep
- “Blocks; low density” ← Elongation

Rule 2 (long “rays”): two blocks, low density (50%), no “camouflage”, no “fame”

U.Pitt, 2/24/17

(c) C. Faloutsos, 2017
Lockstep and Spectral Subspace Plot

- Case #3: non-overlapping lockstep
- “Camouflage” (or “Fame”) \(\rightleftharpoons\) Tilting
- “Rays”

Adjacency Matrix | Spectral Subspace Plot

Rule 3 (tilting “rays”): two blocks, with “camouflage”, no “fame”

U.Pitt, 2/24/17 (c) C. Faloutsos, 2017
Lockstep and Spectral Subspace Plot

- Case #3: non-overlapping lockstep
- “Camouflage” (or “Fame”) ↔ Tilting

“Rays”

Adjacency Matrix  Spectral Subspace Plot

Rule 3 (tilting “rays”): two blocks, no “camouflage”, with “fame”

U.Pitt, 2/24/17  (c) C. Faloutsos, 2017
Lockstep and Spectral Subspace Plot

- Case #4: lockstep
- “?” “Pearls”

Adjacency Matrix

Spectral Subspace Plot
**Lockstep and Spectral Subspace Plot**

- Case #4: **overlapping** lockstep
- “Staircase” → “Pearls”

**Adjacency Matrix**

**Spectral Subspace Plot**

Rule 4 (“pearls”): a “staircase” of three partially overlapping blocks.
Dataset

- Tencent Weibo
- 117 million nodes (with profile and UGC data)
- 3.33 billion directed edges
Real Data

"Pearls"

"Block"

"Rays"

"Staircase"
Real Data

- Spikes on the out-degree distribution
Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
  – P1.1: Patterns
  – P1.2: Anomaly / fraud detection
    • No labels – spectral methods
      – Suspiciousness
    • With labels: Belief Propagation
• Part#2: time-evolving graphs; tensors
• Conclusions
A General Suspiciousness Metric for Dense Blocks in Multimodal Data, Meng Jiang, Alex Beutel, Peng Cui, Bryan Hooi, Shiqiang Yang, and Christos Faloutsos, *ICDM*, 2015.
Suspicious Patterns in Event Data

Which is more suspicious?

20,000 Users
Retweeting same 20 tweets
  6 times each
All in 10 hours

vs.

225 Users
Retweeting same 1 tweet
  15 times each
All in 3 hours
All from 2 IP addresses

U.Pitt, 2/24/17

(c) C. Faloutsos, 2017
Suspicious Patterns in Event Data

Which is more suspicious?

20,000 Users
Retweeting same 20 tweets
6 times each
All in 10 hours

vs.

225 Users
Retweeting same 1 tweet
15 times each
All in 3 hours
All from 2 IP addresses

Answer: volume * $D_{KL}(p \parallel p_{background})$
Suspicious Patterns in Event Data

Which is more suspicious?

20,000 Users
Retweeting same 20 tweets
6 times each
All in 10 hours

225 Users
Retweeting same 1 tweet
15 times each
All in 3 hours
All from 2 IP addresses

Answer: volume * $D_{KL}(p \parallel p_{background})$
Imagie your job at Twitter is to detect when fraudsters are trying to manipulate the most popular tweets for a given region, and sort them in importance ("suspiciousness") order. Our main contribution is that we show how to unify these methods and how to give a principled way to score the suspiciousness of dense blocks indicating fraud, emerging trends, or some other noteworthy insurance fraud, as we discuss next.

Table VI. The performance of CROSSSPOT and HOSVD. The task is again to classify the retweets per user, item, IP, or minute, but does not consider what is boosting and hashtag-hijacking in social datasets spanning regions, and sort them in importance ("suspiciousness") order. We test how the big, dense 225(retweets per user, item, IP, or minute, but does not consider what is boosting and hashtag-hijacking in social datasets spanning regions, and sort them in importance ("suspiciousness") order. We test how the big, dense 225

### CROSSSPOT

<table>
<thead>
<tr>
<th>#</th>
<th>User \times tweet \times IP \times minute</th>
<th>Mass c</th>
<th>Suspiciousness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14 \times 1 \times 2 \times 1,114</td>
<td>41,396</td>
<td>1,239,865</td>
</tr>
<tr>
<td>2</td>
<td>225 \times 1 \times 2 \times 200</td>
<td>27,313</td>
<td>777,781</td>
</tr>
<tr>
<td>3</td>
<td>8 \times 2 \times 4 \times 1,872</td>
<td>17,701</td>
<td>491,323</td>
</tr>
</tbody>
</table>

### HOSVD

<table>
<thead>
<tr>
<th>#</th>
<th>User \times tweet \times IP \times minute</th>
<th>Mass c</th>
<th>Suspiciousness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24 \times 6 \times 11 \times 439</td>
<td>3,582</td>
<td>131,113</td>
</tr>
<tr>
<td>2</td>
<td>18 \times 4 \times 5 \times 223</td>
<td>1,942</td>
<td>74,087</td>
</tr>
<tr>
<td>3</td>
<td>14 \times 2 \times 1 \times 265</td>
<td>9,061</td>
<td>381,211</td>
</tr>
</tbody>
</table>

HOSVD takes only 280 seconds if \( d=2 \) and often even the same text \cite{9}\cite{10}. Zombie followers, botnets reviews for restaurants or hotels will reuse the same accounts always at the same time (when the order for the Page Likes is another example, with (patient-id, doctor-id, prescription-id, hospital-id) traffic is not balanced or the patient is not admitted to the hospital). On the same machine, HOSVD finishes the local search. Each iteration takes only 5.6 seconds. We find that when the number of non-zero entries in the multimodal data is increased by at least 225, the recall value is increased by at least 225.

Summary: CROSSSPOT reports the performances of CROSSSPOT. We observe that in order to find all the six 3-mode injected blocks, it takes 0.979. Figure 3(b) gives the recall value of every testing robustness of the random seed number: We observe that C. Faloutsos, 2017
Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
  – P1.1: Patterns
  – P1.2: Anomaly / fraud detection
    • No labels – spectral methods
    • With labels: 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

Inspected user alisher for suspicious networks.

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
    • No labels - Spectral methods
    • w/ labels: Belief Propagation – closed formulas
• 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)

U.Pitt, 2/24/17
(c) C. Faloutsos, 2017
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>$[I - c AD^{-1}]$</td>
<td>$x$</td>
<td>$(1-c)y$</td>
</tr>
<tr>
<td>SSL</td>
<td>$[I + a(D - A)]$</td>
<td>$x$</td>
<td>$y$</td>
</tr>
<tr>
<td>FABP</td>
<td>$[I + a D - c'A]$</td>
<td>$b_t$</td>
<td>$\phi_t$</td>
</tr>
</tbody>
</table>

### Adjacency Matrix and Final Labels/Beliefs

- **Adjacency Matrix:**
  - $d_1$ to $d_3$,
  - $0 1 0$
  - $1 0 1$
  - $0 1 0$

- **Final Labels/Beliefs:**
  - $?, 0 1$

- **Prior Labels/Beliefs:**
  - $1 1$

---

U.Pitt, 2/24/17       (c) C. Faloutsos, 2017
FABP is linear on the number of edges.
Problem: e-commerce ratings fraud

- Given a heterogeneous graph on users, products, sellers and positive/negative ratings with “seed labels”
- Find the top $k$ most fraudulent users, products and sellers
Problem: e-commerce ratings fraud

- **Given** a heterogeneous graph on users, products, sellers and positive/negative ratings with “seed labels”
- **Find** the top $k$ most fraudulent users, products and sellers

Dhivya Eswaran, Stephan Günnemann, Christos Faloutsos, “ZooBP: Belief Propagation for Heterogeneous Networks”, VLDB 2017
Theorem 1 (ZooBP). If b, e, P, Q are constructed as described above, the linear equation system approximating the final node beliefs given by BP is:

\[ b = e + (P - Q)b \quad (\text{ZooBP}) \]  

Dhivya Eswaran, Stephan Günnemann, Christos Faloutsos, “ZooBP: Belief Propagation for Heterogeneous Networks”, VLDB 2017
ZooBP: features

Fast; convergence guarantees.

Near-perfect accuracy

linear in graph size

Dhivya Eswaran, Stephan Günnemann, Christos Faloutsos, "ZooBP: Belief Propagation for Heterogeneous Networks", VLDB 2017
ZooBP in the real world

- Near 100% precision on top 300 users (Flipkart)
- Flagged users: suspicious
  - 400 ratings in 1 sec
  - 5000 good ratings and no bad ratings

Dhivya Eswaran, Stephan Günnemann, Christos Faloutsos, “ZooBP: Belief Propagation for Heterogeneous Networks”, VLDB 2017
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
  - P2.1: tools/tensors
  - P2.2: other patterns
- 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)
Answer: tensor factorization

- Recall: (SVD) matrix factorization: finds blocks

\[ \mathbf{U} \mathbf{P} \mathbf{U}^T \]

- 'meat-eaters'
- 'steaks'
- 'vegetarians'
- 'plants'
- 'kids'
- 'cookies'

\[ \mathbf{U}_1 \]

\[ \mathbf{V}_1 \]

\[ \mathbf{U}_i \]

\( N \) users

\( M \) products

(c) C. Faloutsos, 2017
Answer: tensor factorization

• Recall: (SVD) matrix factorization: finds blocks
Crush intro to SVD

- Recall: (SVD) matrix factorization: finds blocks

N fans

\[ \sum_{i=1}^{M} \mathbf{u}_i \mathbf{v}_i^T \]

U.Pitt, 2/24/17

(c) C. Faloutsos, 2017
Answer: tensor factorization

- PARAFAC decomposition

\[ \text{subject} \oplus \text{object} = \text{artists} + \text{athletes} + \text{politicians} \]
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!

U.Pitt, 2/24/17
(c) C. Faloutsos, 2017
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!

U.Pitt, 2/24/17
(c) C. Faloutsos, 2017
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
  – P2.1: tools/tensors
  – P2.2: other patterns – inter-arrival time
• Conclusions
KDD 2015 – Sydney, Australia

RSC: Mining and Modeling Temporal Activity in Social Media

Alceu F. Costa*    Yuto Yamaguchi    Agma J. M. Traina
Caetano Traina Jr.    Christos Faloutsos

*alceu.fc@icmc.usp.br
Pattern Mining: Datasets

Reddit Dataset
Time-stamp from comments
21,198 users
20 Million time-stamps

Twitter Dataset
Time-stamp from tweets
6,790 users
16 Million time-stamps

For each user we have:
Sequence of postings time-stamps: \( T = (t_1, t_2, t_3, \ldots) \)
Inter-arrival times (IAT) of postings: \( (\Delta_1, \Delta_2, \Delta_3, \ldots) \)
**Pattern Mining**

**Pattern 1:** Distribution of IAT is heavy-tailed

Users can be inactive for long periods of time before making new postings

IAT Complementary Cumulative Distribution Function (CCDF) (log-log axis)

Reddit Users

Twitter Users

(c) C. Faloutsos, 2017

U.Pitt, 2/24/17
**Pattern Mining**

**Pattern 1:** Distribution of IAT is heavy-tailed

Users can be inactive for long periods of time before making new postings

IAT Complementary Cumulative Distribution Function (CCDF)

No surprises – Should we give up?

Reddit Users

Twitter Users

U.Pitt, 2/24/17

(c) C. Faloutsos, 2017
Human? Robots?

Linear

Log

PDF

Δ, IAT (seconds)

PDF

Δ, IAT (seconds)
Human? Robots?

2’ 3h 1day

linear

log
Experiments: Can RSC-Spotter Detect Bots?

Precision vs. Sensitivity Curves

Good performance: curve close to the top

Twitter

- Precision > 94%
- Sensitivity > 70%

With strongly imbalanced datasets
- # humans >> # bots

(c) C. Faloutsos, 2017
Experiments: Can RSC-Spotter Detect Bots?

Precision vs. Sensitivity Curves

Good performance: curve close to the top

Reddit

Precision > 96%
Sensitivity > 47%
With strongly imbalanced datasets
# humans >> # bots
Roadmap

• Introduction – Motivation
• Part#1: Patterns in graphs
• Part#2: time-evolving graphs
  – P2.1: tools/tensors
  – P2.2: other patterns
    • inter-arrival time
    • Network growth
• Conclusions
Beyond Sigmoids: the NetTide Model for Social Network Growth and its Applications

KDD’16

Chengxi Zang, Peng Cui, CF
PROBLEM: n(t) and e(t), over time?

- n(t): the number of nodes.
- e(t): the number of edges.
- E.g.:
  - How many members will have next month?
  - How many friendship links will have next year?

- Linear?
- Exponential?
- Sigmoid?
Datasets

- **WeChat** 2011/1-2013/1 300M nodes, 4.75B links
- **ArXiv** 1992/3-2002/3 17k nodes, 2.4M links
- **Enron** 1998/1-2002/7 86K nodes, 600K links
- **Weibo** 2006 165K nodes, 331K links
A: Power Law Growth

Cumulative growth (Log-Log scale)
Proposed: NetTide Model

- **Nodes** $n(t)$

  $$
  \frac{dn(t)}{dt} = \frac{\beta}{t^\theta} n(t)(N - n(t))
  $$

- **Links** $e(t)$

  $$
  \frac{de(t)}{dt} = \frac{\beta'}{t^\theta} n(t) \left( \alpha (n(t) - 1)^\gamma - \frac{e(t)}{n(t)} \right) + 2 \frac{dn(t)}{dt}
  $$
NetTide-Node Model

\[
\frac{dn}{dt} = \beta/t^\theta \ n \ (N - n)
\]

- Intuition:
  - Rich-get-richer
  - Limitation
  - Fizzling nature

\{ = Sl; ~Bass
NetTide-Node Model

\[ \frac{dn}{dt} = \beta / t^\theta \times n (N - n) \]

**Intuition:**
- Rich-get-richer
- Limitation
- Fizzling nature

#nodes(t)

Total population

= SI; ~Bass
Results: Accuracy
Results: Accuracy

![Graph showing the growth of WeChat nodes and links over time.](image)

- **WeChat**
  - Nodes
    - Real Node
    - Real Link
    - NT-Node
    - NT-Link
  - #Links
    - Logarithmic scale
    - Real
    - NetTide
  - #Nodes
    - Logarithmic scale
    - Logarithmic scale
  - Slope: 1.41
Results: Accuracy

![Graph 1](left side)

- X-axis: Time (Day)
- Y-axis: No.
- Legend:
  - Real Node
  - Real Link
  - NT-Node
  - NT-Link

![Graph 2](right side)

- X-axis: #Nodes
- Y-axis: #Links
- Legend:
  - Real
  - NetTide

Slope: 1.74
Results: Accuracy

Enron

\[ \text{No.} \quad 10^3 \quad 10^4 \quad 10^5 \quad 10^6 \]
\[ \text{Time (Day)} \quad 750 \quad 1000 \quad 1250 \quad 1500 \]

- Real Node
- Real Link
- NT-Node
- NT-Link

\[ \text{\#Links} \quad 10^4 \quad 10^5 \quad 10^6 \]

- Real
- NetTide

slope: 1.17
Results: Accuracy

[Graphs showing the number of nodes and links over time for the platform Weibo, comparing real and simulated data.]
Results: Forecast

WeChat from 100 million to 300 million

730 days ahead
Part 2: Conclusions

- Time-evolving / heterogeneous graphs -> tensors
- PARAFAC finds patterns
- Surprising temporal patterns (P.L. growth)
Roadmap

• Introduction – Motivation
  – Why study (big) graphs?
• Part#1: Patterns in graphs
• Part#2: time-evolving graphs; tensors
• 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
Cast

Akoglu, Leman
Araujo, Miguel
Beutel, Alex
Chau, Polo
Eswaran, Dhivya
Hooi, Bryan
Kang, U
Koutra, Danai
Papalexakis, Vagelis
Shah, Neil
Shin, Kijung
Song, Hyun Ah

U.Pitt, 2/24/17
(c) C. Faloutsos, 2017
124
CONCLUSION#1 – Big data

• Patterns  Anomalies

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

• powerful tool
References

- D. Chakrabarti, C. Faloutsos: *Graph Mining – Laws, Tools and Case Studies*, Morgan Claypool 2012
TAKE HOME MESSAGE:

Cross-disciplinarity
Thank you!

Cross-disciplinarity