Large Graph Mining – Patterns, Tools and Cascade analysis

Christos Faloutsos
CMU
Thank you!

• Brian Gallagher

• Jan Winfield
Roadmap

• Introduction – Motivation
  – Why ‘big data’
  – Why (big) graphs?
• Problem#1: Patterns in graphs
• Problem#2: Tools
• Problem#3: Scalability
• Conclusions
Why ‘big data’

• Why?
• What is the problem definition?
• What are the major research challenges?
Main message:
Big data: often > experts

• ‘Super Crunchers’ *Why Thinking-By-Numbers is the New Way To Be Smart* by Ian Ayres, 2008

• Google won the machine translation competition 2005
Problem definition – big picture

Tera/Peta-byte data  Analytics  Insights, outliers
Problem definition – big picture

Tera/Peta-byte data

Analytics

Insights, outliers

Main emphasis in this talk
Tera/Peta-byte data

(my personal) rules of thumb: if data
- fits in memory -> R, matlab, scipy
- single disk -> RDBMS (sqlite3, mysql, postgres)
- multiple (<100-1000) disks: parallel RDBMS (Vertica, TeraData)
- multiple (>1000) disks: hadoop, pig
(Free) Resource for graphs

Open source system for mining huge graphs:

PEGASUS project (PEta GrAph mining System)

- www.cs.cmu.edu/~pegasus
- Apache license for s/w
- code and papers
Research challenges

• The usual ones from data mining
  – Data cleansing
  – Feature engineering
  – …

PLUS

  – Scalability ( < O(N**2))
  – Real data *disobey* textbook assumptions
    (uniformity, independence, Gaussian, Poisson)
    with huge performance implications
Roadmap

• Introduction – Motivation
  – Why ‘big data’
  – Why (big) graphs?
• Problem#1: Patterns in graphs
• Problem#2: Tools
• Problem#3: Scalability
• Conclusions
Graphs - why should we care?

> $10B revenue
> >0.5B users

Food Web
[Martinez ’91]

Internet Map
[lumeta.com]
Graphs - why should we care?

- IR: bi-partite graphs (doc-terms)
- web: hyper-text graph
  
- ... and more:
Graphs - why should we care?

- ‘viral’ marketing
- web-log (‘blog’) news propagation
- computer network security: email/IP traffic and anomaly detection
- ....
- Subject-verb-object -> graph
- Many-to-many db relationship -> graph
Outline

• Introduction – Motivation

• Problem#1: Patterns in graphs
  – Static graphs
  – Weighted graphs
  – Time evolving graphs

• Problem#2: Tools

• Problem#3: Scalability

• Conclusions
Problem #1 - network and graph mining

- What does the Internet look like?
- What does FaceBook look like?
- What is ‘normal’/‘abnormal’?
- which patterns/laws hold?
Problem #1 - network and graph mining

- What does the Internet look like?
- What does FaceBook look like?
- What is ‘normal’/‘abnormal’?
- which patterns/laws hold?
  - To spot anomalies (rarities), we have to discover patterns

LLNL, Jan 2013
Problem #1 - network and graph mining

- What does the Internet look like?
- What does FaceBook look like?
- What is ‘normal’/‘abnormal’?
- which patterns/laws hold?
  - To spot anomalies (rarities), we have to discover patterns
  - Large datasets reveal patterns/anomalies that may be invisible otherwise…
Graph mining

• Are real graphs random?
Laws and patterns

• Are real graphs random?
• A: NO!!
  – Diameter
  – 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(degree) vs. log(rank)

---

LLNL, Jan 2013  C. Faloutsos (CMU)  21
Solution# S.1

- Power law in the degree distribution
  [SIGCOMM99]

internet domains

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

-0.82

att.com

ibm.com

LLNL, Jan 2013
C. Faloutsos (CMU)
Solution# S.2: Eigen Exponent $E$

- A2: power law in the eigenvalues of the adjacency matrix

Exponent = slope

$E = -0.48$

May 2001
Solution# S.2: Eigen Exponent $E$

- Eigenvalue

Exponent = slope

$E = -0.48$

- Rank of decreasing eigenvalue

- [Mihail, Papadimitriou ’02]: slope is $\frac{1}{2}$ of rank exponent

May 2001
But:

How about graphs from other domains?
More power laws:

- web hit counts [w/ A. Montgomery]
epinions.com

- who-trusts-whom
  [Richardson + Domingos, KDD 2001]

(out) degree

LLNL, Jan 2013  C. Faloutsos (CMU)
And numerous more

• # of sexual contacts
• Income [Pareto] – ’80-20 distribution’
• Duration of downloads [Bestavros+]
• Duration of UNIX jobs (‘mice and elephants’)
• Size of files of a user
• …
• ‘Black swans’
Roadmap

• Introduction – Motivation

• Problem#1: Patterns in graphs
  – Static graphs
    • degree, diameter, eigen,
    • triangles
    • cliques
  – Weighted graphs
  – Time evolving graphs

• Problem#2: Tools
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?
Triangle Law: #S.3
[Tsourakakis ICDM 2008]

HEP-TH  ASN

Epinions

X-axis: # of participating triangles
Y: count (~ pdf)

LLNL, Jan 2013
Triangle Law: #S.3
[Tsourakakis ICDM 2008]

X-axis: # of participating triangles
Y: count (~ pdf)

HEP-TH
ASN
Epinions

LLNL, Jan 2013
Triangle Law: #S.4
[Tsourakakis ICDM 2008]

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

$n$ friends $\rightarrow \sim n^{1.6}$ triangles

Reuters  \hspace{1cm} SN

Epinions
Triangle Law: Computations

[Tsourakakis ICDM 2008]

But: triangles are expensive to compute
(3-way join; several approx. algos)
Q: Can we do that quickly?
Triangle Law: Computations
[Tsourakakis ICDM 2008]

But: triangles are expensive to compute
(3-way join; several approx. algos)
Q: Can we do that quickly?
A: Yes!
\[
\text{#triangles} = \frac{1}{6} \sum (\lambda_i^3)
\]
(and, because of skewness (S2), we only need the top few eigenvalues!)
Triangle Law: Computations

[Tsourakakis ICDM 2008]

Wikipedia graph 2006-Nov-04
≈ 3.1M nodes ≈ 37M edges

1000x+ speed-up, >90% accuracy
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]
Roadmap

• Introduction – Motivation
• Problem#1: Patterns in graphs
  – Static graphs
    • degree, diameter, eigen,
    • triangles
    • cliques
  – Weighted graphs
  – Time evolving graphs
• Problem#2: Tools
Observations on weighted graphs?

• A: yes - even more ‘laws’!

M. McGlohon, L. Akoglu, and C. Faloutsos
Weighted Graphs and Disconnected Components: Patterns and a Generator.
SIG-KDD 2008
Observation W.1: Fortification

Q: How do the weights of nodes relate to degree?
Observation W.1: Fortification

More donors, more $ ?

$10
$5
$7

‘Reagan’
‘Clinton’
Observation W.1: fortification: Snapshot Power Law

- Weight: super-linear on in-degree
- Exponent ‘iw’: $1.01 < iw < 1.26$

More donors, even more $\$

Orgs-Candidates

E.g. John Kerry, $10M$ received, from $1K$ donors

LLNL, Jan 2013

C. Faloutsos (CMU)
Roadmap

• Introduction – Motivation
• Problem#1: Patterns in graphs
  – Static graphs
  – Weighted graphs
  – Time evolving graphs
• Problem#2: Tools
• ...

LLNL, Jan 2013  C. Faloutsos (CMU)
Problem: Time evolution

• with Jure Leskovec (CMU -> Stanford)

• and Jon Kleinberg (Cornell – sabb. @ CMU)
T.1 Evolution of the Diameter

- Prior work on Power Law graphs hints at **slowly growing diameter**:
  - diameter $\sim O(\log N)$
  - diameter $\sim O(\log \log N)$

- What is happening in real data?
T.1 Evolution of the Diameter

- Prior work on Power Law graphs hints at **slowly growing diameter**:  
  - diameter $\sim O(\log N)$  
  - diameter $\sim O(\log \log N)$

- What is happening in real data?

- Diameter **shrinks** over time
T.1 Diameter – “Patents”

- Patent citation network
- 25 years of data
- @1999
  - 2.9 M nodes
  - 16.5 M edges

![Graph showing the effective diameter over time](image)
T.2 Temporal Evolution of the Graphs

• $N(t)$ … nodes at time $t$
• $E(t)$ … edges at time $t$
• Suppose that
  \[ N(t+1) = 2 \times N(t) \]
• Q: what is your guess for
  \[ E(t+1) =? 2 \times E(t) \]
T.2 Temporal Evolution of the Graphs

• $N(t)$ … nodes at time $t$
• $E(t)$ … edges at time $t$
• Suppose that
  \[ N(t+1) = 2 \times N(t) \]
• Q: what is your guess for
  \[ E(t+1) = \? \times E(t) \]
• A: over-doubled!
  – But obeying the "Densification Power Law"
T.2 Densification – Patent Citations

- Citations among patents granted
- @1999
  - 2.9 M nodes
  - 16.5 M edges
- Each year is a datapoint

\[ N(t) \quad E(t) \]

\[ E(t) = 0.0002 \times 1.66^t \quad R^2 = 0.99 \]

LLNL, Jan 2013  C. Faloutsos (CMU)
Roadmap

• Introduction – Motivation
• Problem#1: Patterns in graphs
  – Static graphs
  – Weighted graphs
  – Time evolving graphs
• Problem#2: Tools
• ...

LLNL, Jan 2013  C. Faloutsos (CMU)
T.3 : popularity over time

Post popularity drops-off – exponentially?

LLNL, Jan 2013

C. Faloutsos (CMU)
T.3: popularity over time

Post popularity drops-off — exponentially?  
POWER LAW!  
Exponent?
T.3 : popularity over time

Post popularity drops-off – exponentially?  
POWER LAW!
Exponent? -1.6
- close to -1.5: Barabasi’s stack model
- and like the zero-crossings of a random walk

LLNL, Jan 2013  
C. Faloutsos (CMU)
-1.5 slope


\[
\text{Prob}(RT > x) \quad (\log)
\]

Response time (log)
Roadmap

• Introduction – Motivation
• Problem#1: Patterns in graphs
• Problem#2: Tools
  – Belief Propagation
  – Tensors
  – Spike analysis
• Problem#3: Scalability
• 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

Suspected fraudster – this user has been behaving much like the other suspects by frauding with the similar sets of possible accomplices.
E-bay Fraud detection - NetProbe

Compatibility matrix

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>A</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>99%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>99%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>49%</td>
<td>49%</td>
<td></td>
</tr>
</tbody>
</table>

heterophily
Roadmap

• Introduction – Motivation
• Problem#1: Patterns in graphs
• Problem#2: Tools
  – Belief Propagation
  – Tensors
  – Spike analysis
• Problem#3: Scalability
• Conclusions
GigaTensor: Scaling Tensor Analysis Up By 100 Times – Algorithms and Discoveries

U Kang
Evangelos Papalexakis
Abhay Harpale
Christos Faloutsos

KDD’12
Background: Tensor

- Tensors (=multi-dimensional arrays) are everywhere
  - Hyperlinks & anchor text [Kolda+, 05]
Background: Tensor

- Tensors (=multi-dimensional arrays) are everywhere
  - Sensor stream (time, location, type)
  - Predicates (subject, verb, object) in knowledge base

“Eric Clapton plays guitar”

“Barrack Obama is the president of U.S.”

NELL (Never Ending Language Learner) data
Nonzeros = 144M

LLNL, Jan 2013

C. Faloutsos (CMU)
Background: Tensor

- Tensors (=multi-dimensional arrays) are everywhere
  - Sensor stream (time, location, type)
  - Predicates (subject, verb, object) in knowledge base
all I learned on tensors: from

Nikos Sidiropoulos
UMN

Tamara Kolda,
Sandia Labs
(tensor toolbox)
Problem Definition

• How to decompose a billion-scale tensor?
  – Corresponds to SVD in 2D case
Problem Definition

- Q1: Dominant concepts/topics?
- Q2: Find synonyms to a given noun phrase?
- (and how to scale up: $|\text{data}| > \text{RAM}$)

\( (48M) \text{ verbs} \)

\( (26M) \text{ subjects} \)

\( (26M) \text{ objects} \)

NELL (Never Ending Language Learner) data
Nonzeros = 144M

LLNL, Jan 2013

C. Faloutsos (CMU)
Experiments

• GigaTensor solves \(100x\) larger problem

\[
\text{Number of nonzero} = \frac{I}{50}
\]

\((K)\)
\((J)\)
\((I)\)

Number of nonzero
\(= \frac{I}{50}\)
A1: Concept Discovery

- Concept Discovery in Knowledge Base

| Concept 1: "Web Protocol" | Concept 2: "Credit Cards" | Concept 3: "Health System" | Concept 4: "Family Life"
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>internet</td>
<td>credit</td>
<td>health</td>
<td>life</td>
</tr>
<tr>
<td>protocol</td>
<td>information</td>
<td>provider</td>
<td>rest</td>
</tr>
<tr>
<td>file</td>
<td>debt</td>
<td>providers</td>
<td>part</td>
</tr>
<tr>
<td>software</td>
<td>library</td>
<td>system</td>
<td>part</td>
</tr>
<tr>
<td>data</td>
<td>number</td>
<td></td>
<td>years</td>
</tr>
<tr>
<td>suite</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>‘np1’ ‘marketing’ ‘np2’</td>
<td>‘np1’ ‘report’ ‘np2’</td>
<td>‘np1’ ‘insurance’ ‘np2’</td>
<td>‘np2’ ‘of’ ‘his’ ‘np1’</td>
</tr>
<tr>
<td>‘np1’ ‘dating’ ‘np2’</td>
<td>‘np1’ ‘cards’ ‘np2’</td>
<td>‘np1’ ‘service’ ‘np2’</td>
<td>‘np2’ ‘of’ ‘her’ ‘np1’</td>
</tr>
</tbody>
</table>
## A1: Concept Discovery

<table>
<thead>
<tr>
<th>Noun Phrase 1</th>
<th>Noun Phrase 2</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>internet</td>
<td>protocol</td>
<td>‘np1’ ‘stream’ ‘np2’</td>
</tr>
<tr>
<td>file</td>
<td>software</td>
<td>‘np1’ ‘marketing’ ‘np2’</td>
</tr>
<tr>
<td>data</td>
<td>suite</td>
<td>‘np1’ ‘dating’ ‘np2’</td>
</tr>
</tbody>
</table>

**Concept 1: "Web Protocol"**

**Concept 2: "Credit Cards"**

<table>
<thead>
<tr>
<th>Credit</th>
<th>information</th>
<th>‘np1’ ‘card’ ‘np2’</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit</td>
<td>debt</td>
<td>‘np1’ ‘report’ ‘np2’</td>
</tr>
<tr>
<td>library</td>
<td>number</td>
<td>‘np1’ ‘cards’ ‘np2’</td>
</tr>
</tbody>
</table>

**Concept 3: "Health System"**

<table>
<thead>
<tr>
<th>health</th>
<th>provider</th>
<th>‘np1’ ‘care’ ‘np2’</th>
</tr>
</thead>
<tbody>
<tr>
<td>child</td>
<td>providers</td>
<td>‘np’ ‘insurance’ ‘np2’</td>
</tr>
<tr>
<td>home</td>
<td>system</td>
<td>‘np1’ ‘service’ ‘np2’</td>
</tr>
</tbody>
</table>
## A2: Synonym Discovery

<table>
<thead>
<tr>
<th>(Given) Noun Phrase</th>
<th>(Discovered) Potential Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>pollutants</td>
<td>dioxin, sulfur dioxide,</td>
</tr>
<tr>
<td></td>
<td>greenhouse gases, particulates,</td>
</tr>
<tr>
<td></td>
<td>nitrogen oxide, air pollutants,</td>
</tr>
<tr>
<td></td>
<td>cholesterol</td>
</tr>
<tr>
<td>disabilities</td>
<td>infections, dizziness,</td>
</tr>
<tr>
<td></td>
<td>injuries, diseases, drowsiness,</td>
</tr>
<tr>
<td></td>
<td>stiffness, injuries</td>
</tr>
<tr>
<td>vodafone</td>
<td>verizon, comcast</td>
</tr>
<tr>
<td>Christian history</td>
<td>European history, American history,</td>
</tr>
<tr>
<td></td>
<td>Islamic history, history</td>
</tr>
<tr>
<td>disbelief</td>
<td>dismay, disgust, astonishment</td>
</tr>
</tbody>
</table>
Roadmap

• Introduction – Motivation
• Problem#1: Patterns in graphs
• Problem#2: Tools
  – Belief propagation
  – Tensors
  – Spike analysis
• Problem#3: Scalability -PEGASUS
• Conclusions
Rise and fall patterns in social media

• Meme (# of mentions in blogs)
  – short phrases Sourced from U.S. politics in 2008

“you can put lipstick on a pig”

“yes we can”
Rise and fall patterns in social media

• Can we find a unifying model, which includes these patterns?
  • **four** classes on YouTube [Crane et al. ’08]
  • **six** classes on Meme [Yang et al. ’11]
Rise and fall patterns in social media

• Answer: YES!

We can represent all patterns by single model

In Matsubara+ SIGKDD 2012
Main idea - SpikeM

- 1. Un-informed bloggers (uninformed about rumor)
- 2. External shock at time $n_b$ (e.g., breaking news)
- 3. Infection (word-of-mouth)

Infectiveness of a blog-post at age $n$:

- $\beta$ – Strength of infection (quality of news)
- $f(n)$ – Decay function
Main idea - SpikeM

- 1. Un-informed bloggers (uninformed about rumor)
- 2. External shock at time $n_b$ (e.g., breaking news)
- 3. Infection (word-of-mouth)

Infectiveness of a blog-post at age $n$:

- Strength of infection (quality of news): $\beta$
- Decay function: $f(n) = \beta * n^{-1.5}$
SpikeM - with periodicity

- Full equation of SpikeM

\[ \Delta B(n+1) = p(n+1) \cdot \left[ U(n) \cdot \sum_{t=n_b}^{n} (\Delta B(t) + S(t)) \cdot f(n+1 - t) + \varepsilon \right] \]

Bloggers change their activity over time (e.g., daily, weekly, yearly)
Details

- Analysis – exponential rise and power-law fall

Lin-log

Log-log

Rise-part

SI -> exponential
SpikeM -> exponential
Details

- Analysis – exponential rise and power-law fall

- Fall-part
  - SI \(\rightarrow\) exponential
  - SpikeM \(\rightarrow\) power law
Tail-part forecasts

- **SpikeM** can capture tail part

![Diagram showing tail-part forecasts](image)

- $N = 5960, \beta N = 0.7$
- $N = 3481, \beta N = 1.2$
“What-if” forecasting

(1) First spike  (2) Release date  (3) Two weeks before release

- (1) First spike
- (2) Release date
- (3) Two weeks before release

- November 19, 2010 "Deathly Hallows part 1"
- July 15, 2009 "Harry Potter and the Half-Blood Prince"
- July 15, 2011 "Deathly Hallows part 2"

e.g., given (1) first spike,
(2) release date of two sequel movies
(3) access volume before the release date
“What-if” forecasting

(1) First spike
(2) Release date
(3) Two weeks before release

November 19, 2010
"Deathly Hallows part 1"

July 15, 2009
"Harry Potter and the Half-Blood Prince"

July 15, 2011
"Deathly Hallows part 2"

SpikeM can forecast upcoming spikes
Roadmap

• Introduction – Motivation
• Problem#1: Patterns in graphs
• Problem#2: Tools
• Problem#3: Scalability – PEGASUS
  – Diameter
  – Connected components
• Conclusions
Scalability

- Yahoo: 5Pb of data [Fayyad, KDD’07]
- Problem: machine failures, on a daily basis
- How to parallelize data mining tasks, then?
## Roadmap – Algorithms & results

<table>
<thead>
<tr>
<th></th>
<th>Centralized</th>
<th>Hadoop/PEGASUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree Distr.</td>
<td>old</td>
<td>old</td>
</tr>
<tr>
<td>Pagerank</td>
<td>old</td>
<td>old</td>
</tr>
<tr>
<td>Diameter/ANF</td>
<td>old</td>
<td><strong>HERE</strong></td>
</tr>
<tr>
<td>Conn. Comp</td>
<td>old</td>
<td><strong>HERE</strong></td>
</tr>
<tr>
<td>Triangles</td>
<td><strong>done</strong></td>
<td><strong>HERE</strong></td>
</tr>
<tr>
<td>Visualization</td>
<td><strong>started</strong></td>
<td></td>
</tr>
</tbody>
</table>
HADI for diameter estimation

- *Radius Plots for Mining Tera-byte Scale Graphs* U Kang, Charalampos Tsourakakakis, Ana Paula Appel, Christos Faloutsos, Jure Leskovec, SDM’10

- Naively: diameter needs $O(N^{**2})$ space and up to $O(N^{**3})$ time – prohibitive ($N \sim 1B$)

- Our HADI: linear on $E$ ($\sim 10B$)
  - Near-linear scalability wrt # machines
  - Several optimizations -> 5x faster
19+ [Barabasi+]

~1999, ~1M nodes

Number of Nodes

0 5 10 15 20 25 30

Radius

Count

LLNL, Jan 2013

C. Faloutsos (CMU)
YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)
- Largest publicly available graph ever studied.
YahooWeb graph  (120Gb, 1.4B nodes, 6.6 B edges)
• Largest publicly available graph ever studied.
YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)
• 7 degrees of separation (!)
• Diameter: shrunk

14 (dir.)

~7 (undir.)

19+? [Barabasi+]
CMU SCS

YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

Q: Shape?

~7 (undir.)
YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)
• effective diameter: surprisingly small.
• Multi-modality (?!)

LLNL, Jan 2013  C. Faloutsos (CMU)
Radius Plot of **GCC** of YahooWeb.
YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)
• effective diameter: surprisingly small.
• Multi-modality: probably mixture of cores.
YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)
• effective diameter: surprisingly small.
• Multi-modality: probably mixture of cores.

Conjecture:
EN
~7
DE
BR

LLNL, Jan 2013
C. Faloutsos (CMU)
YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)
- effective diameter: surprisingly small.
- Multi-modality: probably mixture of cores.

Conjecture:
Running time - Kronecker and Erdos-Renyi Graphs with billions edges.
Roadmap

• Introduction – Motivation
• Problem#1: Patterns in graphs
• Problem#2: Tools
• Problem#3: Scalability – PEGASUS
  – Diameter
  – Connected components
• Conclusions
Generalized Iterated Matrix Vector Multiplication (GIMV)

**PEGASUS: A Peta-Scale Graph Mining System - Implementation and Observations.**
U Kang, Charalampos E. Tsourakakis, and Christos Faloutsos.
Generalized Iterated Matrix Vector Multiplication (GIMV)

- PageRank
- proximity (RWR)
- Diameter
- Connected components
  - (eigenvectors,
    - Belief Prop.
  - … )

Matrix – vector Multiplication (iterated)
Example: GIM-V At Work

• Connected Components – 4 observations:
Example: GIM-V At Work

• Connected Components

Count

Size

1) 10K x larger than next
Example: GIM-V At Work

- Connected Components

2) ~0.7B singleton nodes
Example: GIM-V At Work

- Connected Components

3) SLOPE!
Example: GIM-V At Work

- Connected Components

4) Spikes!
Example: GIM-V At Work

• Connected Components

Count

Size

suspicious financial-advice sites (not existing now)
GIM-V At Work

• Connected Components over Time
• LinkedIn: 7.5M nodes and 58M edges

Stable tail slope after the gelling point
Roadmap

• Introduction – Motivation
• Problem#1: Patterns in graphs
• Problem#2: Tools
• Problem#3: Scalability
• Conclusions
OVERALL CONCLUSIONS – low level:

• Several new **patterns** (fortification, triangle-laws, conn. components, etc)

• **New tools:**
  – belief propagation, gigaTensor, etc

• **Scalability:** PEGASUS / hadoop
OVERALL CONCLUSIONS –
high level

- **BIG DATA**: Large datasets reveal patterns/outliers that are invisible otherwise

---

![Graph showing multi-modal distribution with effective diameter at 7.62](image-url)
References

• Leman Akoglu, Christos Faloutsos: *RTG: A Recursive Realistic Graph Generator Using Random Typing*. ECML/PKDD (1) 2009: 13-28

References


References

• Christos Faloutsos, Tamara G. Kolda, Jimeng Sun: *Mining large graphs and streams using matrix and tensor tools*. Tutorial, SIGMOD Conference 2007: 1174
References

References


References

• Yasuko Matsubara, Yasushi Sakurai, B. Aditya Prakash, Lei Li, Christos Faloutsos, "Rise and Fall Patterns of Information Diffusion: Model and Implications", KDD’12, pp. 6-14, Beijing, China, August 2012
References

• Jimeng Sun, Yinglian Xie, Hui Zhang, Christos Faloutsos. Less is More: Compact Matrix Decomposition for Large Sparse Graphs, SDM, Minneapolis, Minnesota, Apr 2007.

• Jimeng Sun, Spiros Papadimitriou, Philip S. Yu, and Christos Faloutsos, GraphScope: Parameter-free Mining of Large Time-evolving Graphs ACM SIGKDD Conference, San Jose, CA, August 2007
References

• Jimeng Sun, Dacheng Tao, Christos Faloutsos: *Beyond streams and graphs: dynamic tensor analysis*. KDD 2006: 374-383
References


• Hanghang Tong, Christos Faloutsos, *Center-Piece Subgraphs: Problem Definition and Fast Solutions*, KDD 2006, Philadelphia, PA
References

• Hanghang Tong, Christos Faloutsos, Brian Gallagher, Tina Eliassi-Rad: *Fast best-effort pattern matching in large attributed graphs*. KDD 2007: 737-746
Project info & ‘thanks’

www.cs.cmu.edu/~pegasus

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

Akoglu, Leman
Beutel, Alex
Chau, Polo
Kang, U
Koutra, Danae

McGlohon, Mary
Prakash, Aditya
Papalexakis, Vagelis
Tong, Hanghang

LLNL, Jan 2013

C. Faloutsos (CMU)

130
Thanks to LLNL colleagues

Brian Gallagher

Keith Henderson
Take-home message

Big data reveal \textit{insights} that would be invisible otherwise (even to \textit{experts}).