Mining Large Graphs: Patterns, Anomalies, and Fraud Detection

Christos Faloutsos
CMU
Thank you!

- Nina Balcan

- Kilian Weinberger
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?

>$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

Patterns anomalies
destination
source time
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

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

- att.com
- ibm.com
Solution S.1

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

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

internet domains

- att.com
- ibm.com
Solution# S.1

- Q: So what?

**internet domains**

- **att.com**
- **ibm.com**

$\log(\text{rank}) \sim -0.82 \log(\text{degree})$

ICML'16

(c) 2016, C. Faloutsos
Q: So what?

A1: # of two-step-away pairs: internet domains

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

$\text{att.com}$

$\text{ibm.com}$

$\exp(6.55663) \times x^{(-0.82 \pm 0.16)}$
Solution# S.1

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

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

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

ICML'16
(c) 2016, C. Faloutsos
Solution# S.1

• Q: So what?

• A1: # of two-step-away pairs: $100^2 \times N = 10$ Trillion internet domains

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

log(rank)

log(degree)

-0.82

att.com

ibm.com
Solution # S.1

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

Gaussian trap

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

~0.8PB -> a data center(!)

internet domains

att.com

ibm.com

log(degree)

log(rank)

-0.82

DCO @ CMU

(c) 2016, C. Faloutsos
Solution# S.1

- Q: So what?
- A1: # of two-step-away internet

\[ O(d_{max}^2) \sim 10M^2 \log(\text{rank}) \log(\text{degree}) \]

\[ \sim 0.8PB \rightarrow \text{a data center(!)} \]

\[ \sim 0.8 \text{PB} \rightarrow \text{a data center(!)} \]

Such patterns -> New algorithms

Gaussian trap
Observation – big-data:

• $O(N^2)$ algorithms are \textit{~}intractable - $N=1$B

• $N^2$ seconds = 31B years (>2x age of universe)
Observation – big-data:

- \( O(N^2) \) algorithms are \( \sim \) intractable - \( N = 1B \)
- \( N^2 \) seconds = 31B years
- 1,000 machines
Observation – big-data:

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

• $N^2$ seconds = 31B years

• 1M machines
Observation – big-data:

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

• $N^2$ seconds = 31B years

• 10B machines ~ $10\text{Trillion}$
Observation – big-data:

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

And parallelism might not help

- $N^2$ seconds = 31B years
- 10B machines $\sim$ $10$Trillion

ICML'16

(c) 2016, C. Faloutsos
Solution# S.2: Eigen Exponent $E$

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

• Introduction – Motivation
• Part#1: Patterns in graphs
  – Patterns: Degree; Triangles
  – Anomaly/fraud detection
  – Graph understanding
• 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]

Reuters

Epinions

X-axis: degree
Y-axis: mean # triangles
n friends -> \( \sim n^{1.6} \) triangles

ICML'16
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) – $O(d_{\text{max}}^2)$

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]
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.</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>✔️ L04.</td>
<td>Community structure [Flake et al. ‘02, Girvan and Newman ‘02]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic</th>
<th>Unweighted</th>
<th>Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>L06.</td>
<td>Small and shrinking diameter [Albert and Barabási ‘99, Leskovec et al. ‘05]</td>
<td></td>
</tr>
<tr>
<td>L07.</td>
<td>Constant size 2nd and 3rd connected components [McGlohon et al. ‘08]</td>
<td></td>
</tr>
<tr>
<td>L08.</td>
<td>Principal Eigenvalue Power Law ($\lambda_1$PL) [Akoglu et al. ‘08]</td>
<td></td>
</tr>
<tr>
<td>L09.</td>
<td>Bursty/self-similar edge/weight additions [Gomez and Santona ‘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>L02. Triangle Power Law (TPL) [Tsourakakis ’08]</td>
<td>L09. Weight Power Law (WPL) [McGlohon et al. ’08]</td>
</tr>
<tr>
<td>L06. Small and shrinking diameter [Albert and Barabasi ’99, Leskovec et al. ’05]</td>
<td>L07. Constant size 2^m and 3^m connected components [McGlohon et al. ’08]</td>
</tr>
</tbody>
</table>


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

**CopyCatch: Stopping Group Attacks by Spotting Lockstep Behavior in Social Networks**, Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, Christos Faloutsos *WWW, 2013.*
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

(c) 2016, C. Faloutsos
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
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 number of users caught over time](image)
Deployment at Facebook

Manually labeled 22 randomly selected clusters from February 2013
Deployment at Facebook

Fake acct

- Fake Accounts
- Malicious Browser Extensions
- OS Malware
- Credential Stealing
- Social Engineering

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

Manually labeled 22 randomly selected clusters from February 2013
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

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’, suspiciousness)
    • Belief Propagation

• Part#2: time-evolving graphs; tensors

• Conclusions
Suspicious Patterns in Event Data

2-modes

\[ \text{?} \quad \text{?} \]

\[ \text{?} \quad \text{?} \]

\[ \text{?} \quad \text{?} \]

n-modes

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

Answer: volume * $D_{KL}(p||p_{background})$
Suspicous Patterns in Event Data

![Image of a network graph showing 200 minutes of retweeting activity with 225 users and 27,313 retweets.]

Retweeting: “Galaxy Note Dream Project: Happy Happy Life Traveling the World”

<table>
<thead>
<tr>
<th></th>
<th>User × tweet × IP × minute</th>
<th>Mass</th>
<th>Suspiciousness</th>
</tr>
</thead>
<tbody>
<tr>
<td>CROSSSPOT</td>
<td>14 × 1 × 2 × 114</td>
<td>41,396</td>
<td>1,239,865</td>
</tr>
<tr>
<td></td>
<td>225 × 1 × 2 × 200</td>
<td>27,313</td>
<td>777,781</td>
</tr>
<tr>
<td></td>
<td>8 × 2 × 4 × 1,872</td>
<td>17,701</td>
<td>491,323</td>
</tr>
<tr>
<td>1</td>
<td>24 × 6 × 11 × 439</td>
<td>3,582</td>
<td>131,113</td>
</tr>
<tr>
<td>2</td>
<td>18 × 4 × 5 × 223</td>
<td>1,942</td>
<td>74,087</td>
</tr>
<tr>
<td>3</td>
<td>14 × 2 × 1 × 265</td>
<td>9,061</td>
<td>381,211</td>
</tr>
</tbody>
</table>

(C) 2016, C. Faloutsos
Roadmap

- **Introduction – Motivation**
- **Part#1: Patterns in graphs**
  - Patterns
  - Anomaly / fraud detection
    - CopyCatch
    - Spectral methods (‘fBox’)
    - (Matrix re-ordering + education -> ‘groupNteach’)
    - Belief Propagation
- **Part#2: time-evolving graphs; tensors**
- **Conclusions**
Problem dfn:

e.g.
Problem definition

• **Given** a large binary matrix of facts of 
  *(object, property)* pairs

• **Find** *groupings* of the facts and the *order* of transmission

• To **optimize** ‘student effort’ (→ incremental learning curve, ‘*ALOC*’)

Given a large binary matrix of objects and properties, re-order rows and columns,

**G1. Metric** for better encoding of matrix for student learning?  
**G2. How do we construct language to describe it?**  
**G3. How do we optimize this metric?**
Pictorial Problem definition

ALOC vs. #bits transmitted

#dots learned

Score
Results

Drugs → Side effects

Anti-depressants → Teaching order → Hyper-tension → Pain
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: Unearth Networks of Suspicious Auction Users

Inspect user: alisher for suspicious networks.

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

Registration: Aug 13, 06; Location: United States

Fraudsters: 95% Accountability: 1%

NetProbe diagram showing network connections between users.
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>$[I - c \ A \ D^{-1}]$</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>$b_h$</td>
<td>$\phi_h$</td>
</tr>
</tbody>
</table>

### Adjacency Matrix

```
1 1 1
1 1 1
d1 d2 d3
0 1 0
1 0 1
0 1 0
```

### Final Labels/Beliefs

```
? 
```

### Prior Labels/Beliefs

```
0 1 1
```
Results: Scalability

FABP is linear on the number of edges.
Summary of Part#1

• *many* patterns in real graphs
  – Power-laws everywhere
  – Gaussian trap
    • Avg << Max
  – Long (and growing) list of tools for anomaly/fraud detection

Patterns \times 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]
- 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
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'
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

\[ \text{~} + \quad \text{~} + \quad \text{~} \]

\[ \mathbf{u}_1 \quad \mathbf{v}_1 \quad \mathbf{w}_i \]

\[ N \quad M \]

\[ \text{users} \quad \text{products} \]
Answer to both: tensor factorization

- PARAFAC decomposition

\[ \text{subject} \times \text{verb} \times \text{object} = \text{politicans} + \text{artists} + \text{athletes} \]
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!
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?
**Brain Scan Data**
- 9 persons
- 60 nouns

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

**Patterns?**
- airplane
- dog
Nouns Questions

Nouns
- beetle
- pants
- bee

Questions
- can it cause you pain?
- do you see it daily?
- is it conscious?

Nouns
- bear
- cow
- coat

Questions
- does it grow?
- is it alive?
- was it ever alive?

Nouns
- glass
- tomato
- bell

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

Nouns
- bed
- house
- car

Questions
- does it use electricity?
- can you sit on it?
- does it cast a shadow?

Figure 4: Neuro-semantics
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?”
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
Part 2: Conclusions

• Time-evolving / heterogeneous graphs -> tensors
• PARAFAC finds patterns
• (GigaTensor/HaTen2 -> fast & scalable)
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
CONCLUSION#1 – Big data

- Patterns ≠ Anomalies

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

• powerful tool

1 caller

5 receivers

4 days of activity
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
Thank you!

Cross-disciplinarity

http://www.cs.cmu.edu/~christos/TALKS/16-06-19-ICML/