Mining Large Dynamic Graphs and Tensors

Kijung Shin

Ph.D. Student

Carnegie Mellon University

(kijungs@cs.cmu.edu)

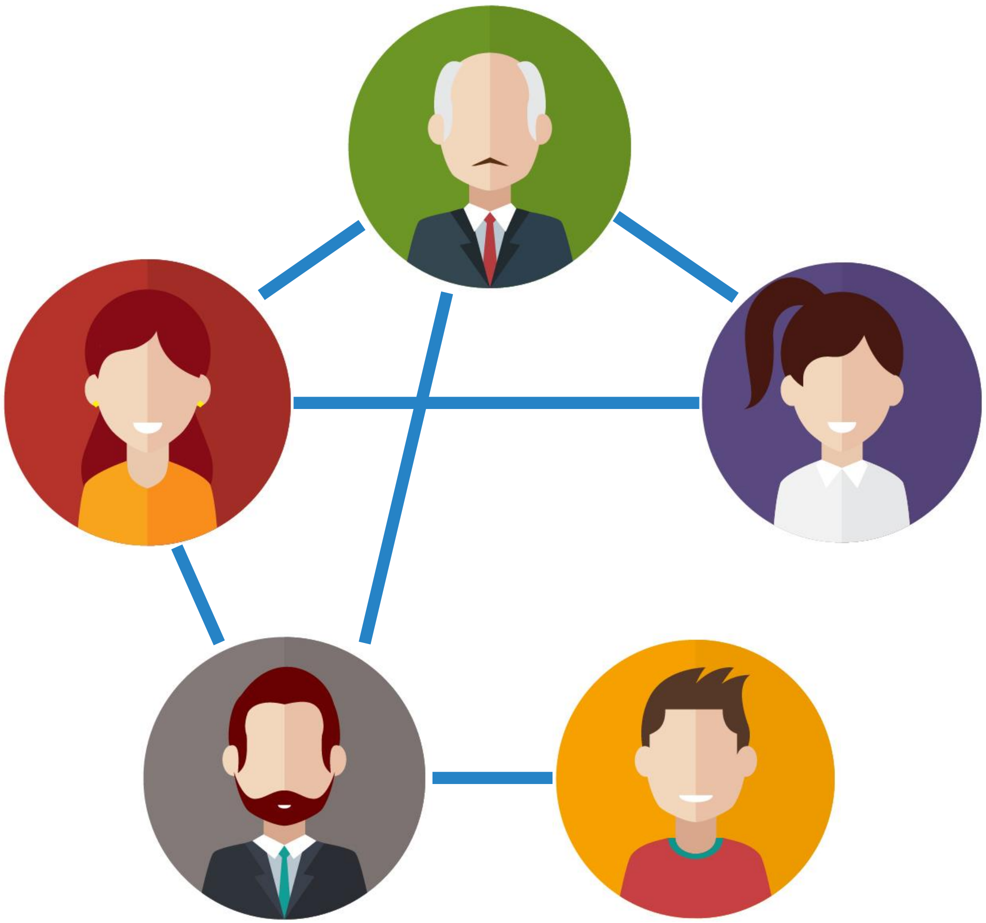
Thesis Committee

- Prof. Christos Faloutsos (Chair)
- Prof. Tom M. Mitchell
- Prof. Leman Akoglu
- Prof. Philip S. Yu

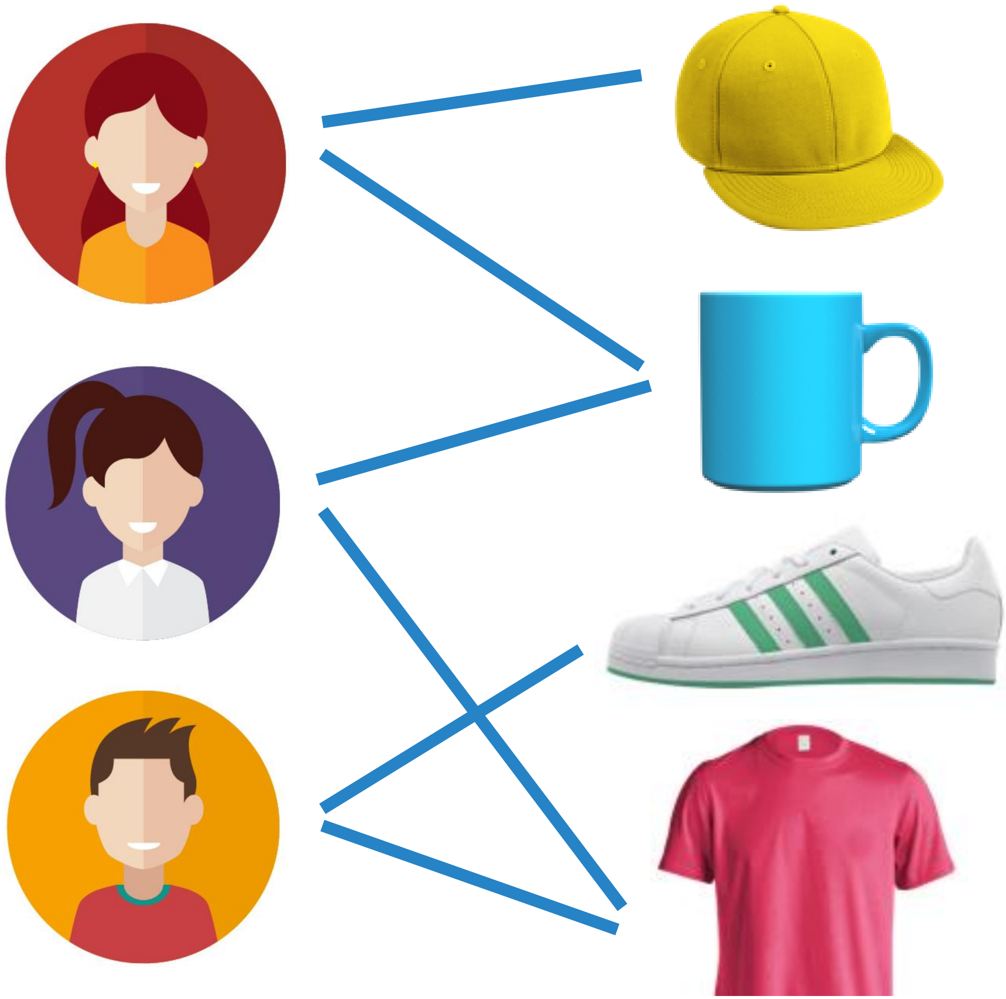



Mining Large Dynamic Graphs and Tensors

Graphs: Social Networks

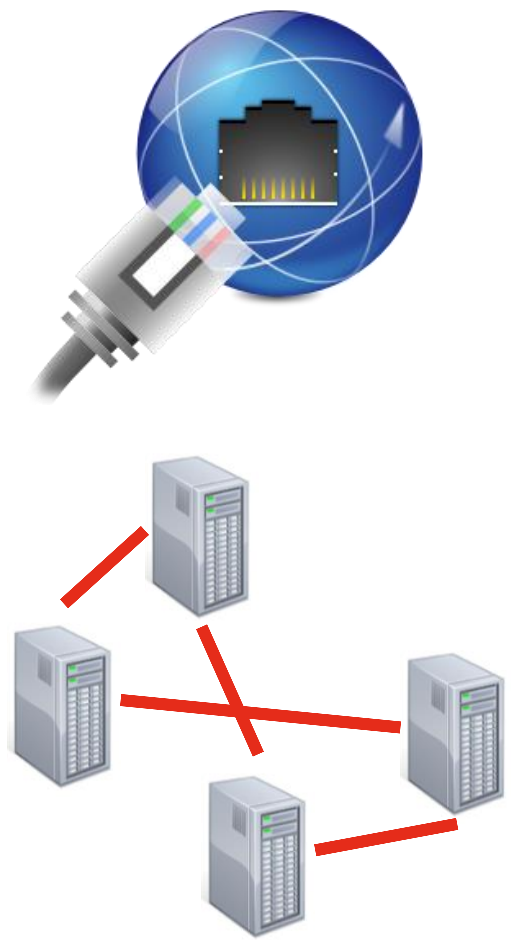


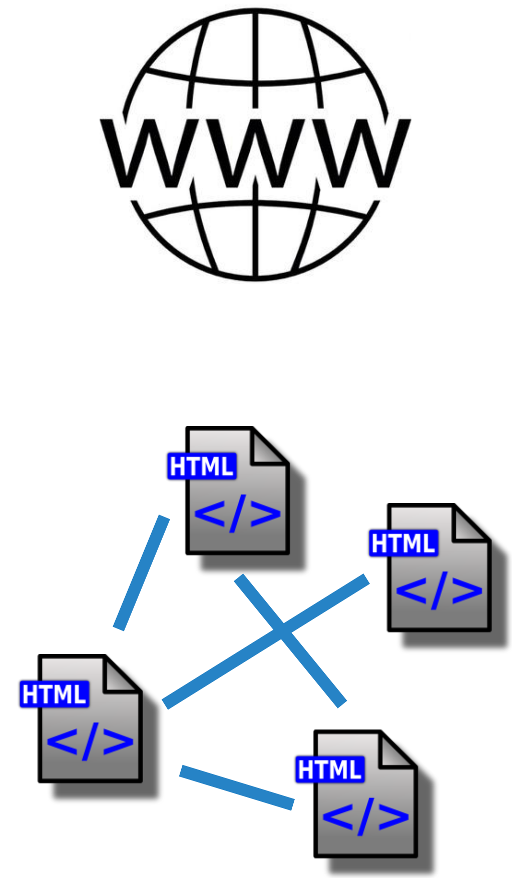
Graphs: Purchase History

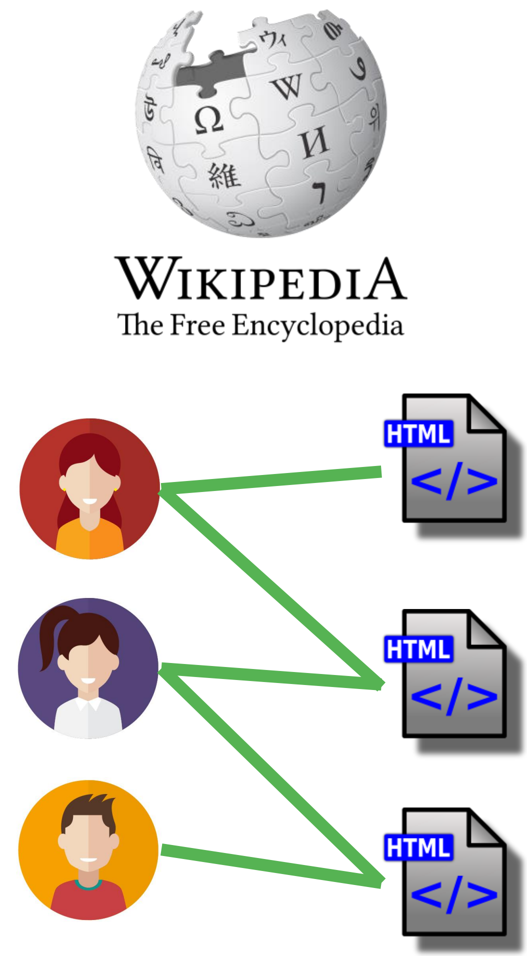


amazon
ebay
Alibaba.com

Graphs: Many More







Properties of Real-world Graphs

- **Large**: many nodes, more edges

40B+ web pages

2B+ active users

500M+ products

5M+ articles

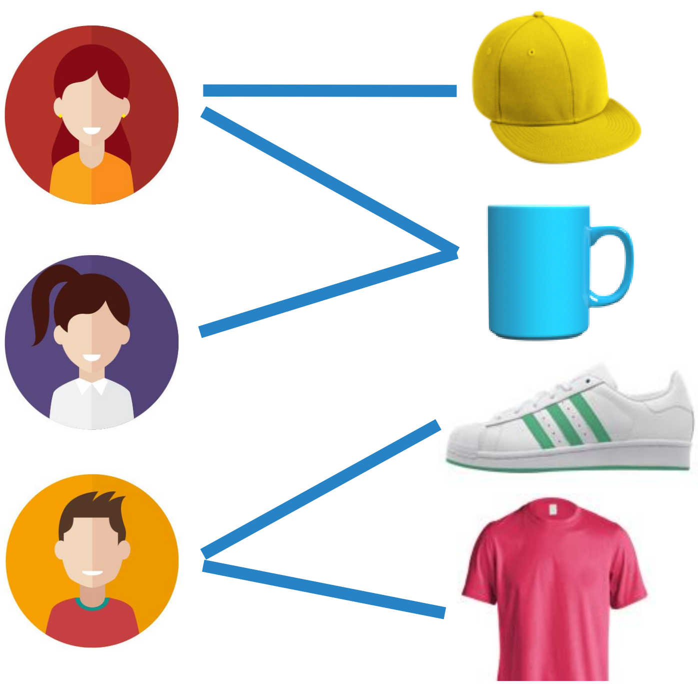
- **Dynamic**: additions/deletions of nodes and edges

Properties of Real-world Graphs

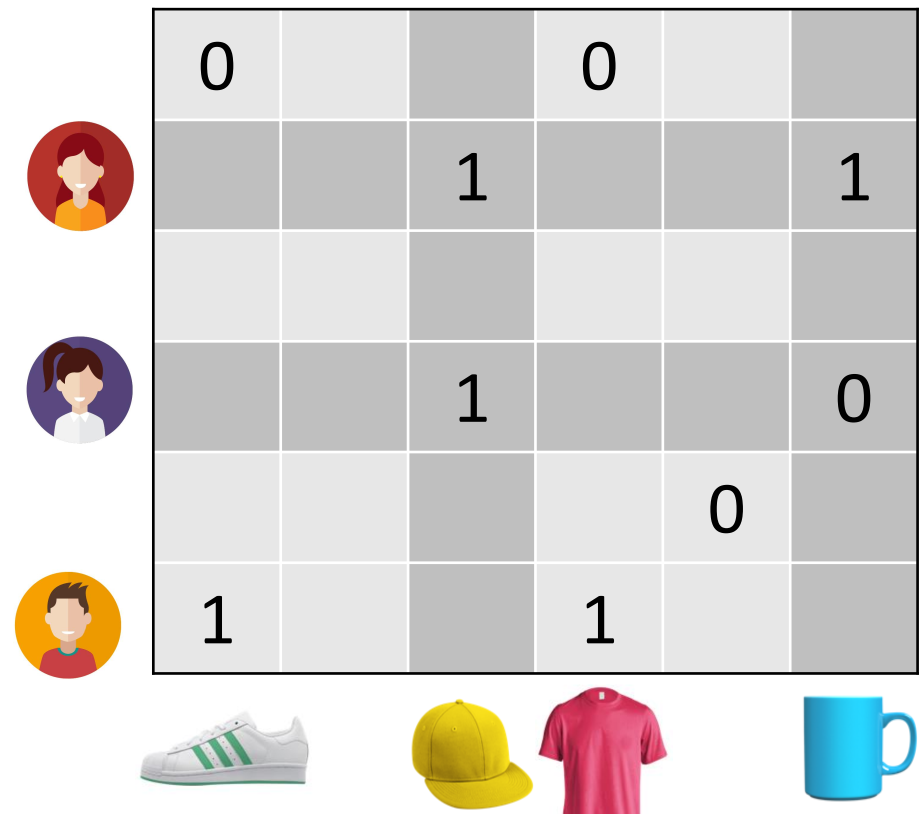
- **Rich with Attributes**: timestamps, scores, text, etc.

Matrices for Graphs

Graph



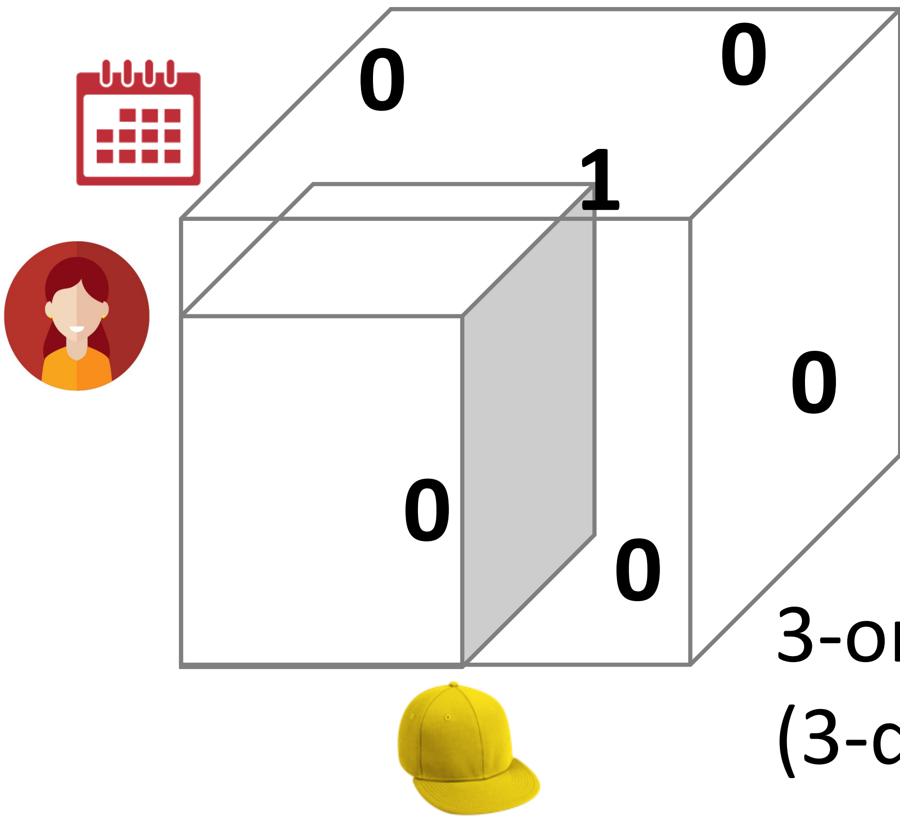
Adjacency Matrix



0			0		
		1			1
		1			0
				0	
1			1		

Tensors for Rich Graphs

- **Tensors**: multi-dimensional array



+ Stars
(4-order tensor)

+ Text
(5-order tensor)

3-order tensor
(3-dimensional array)

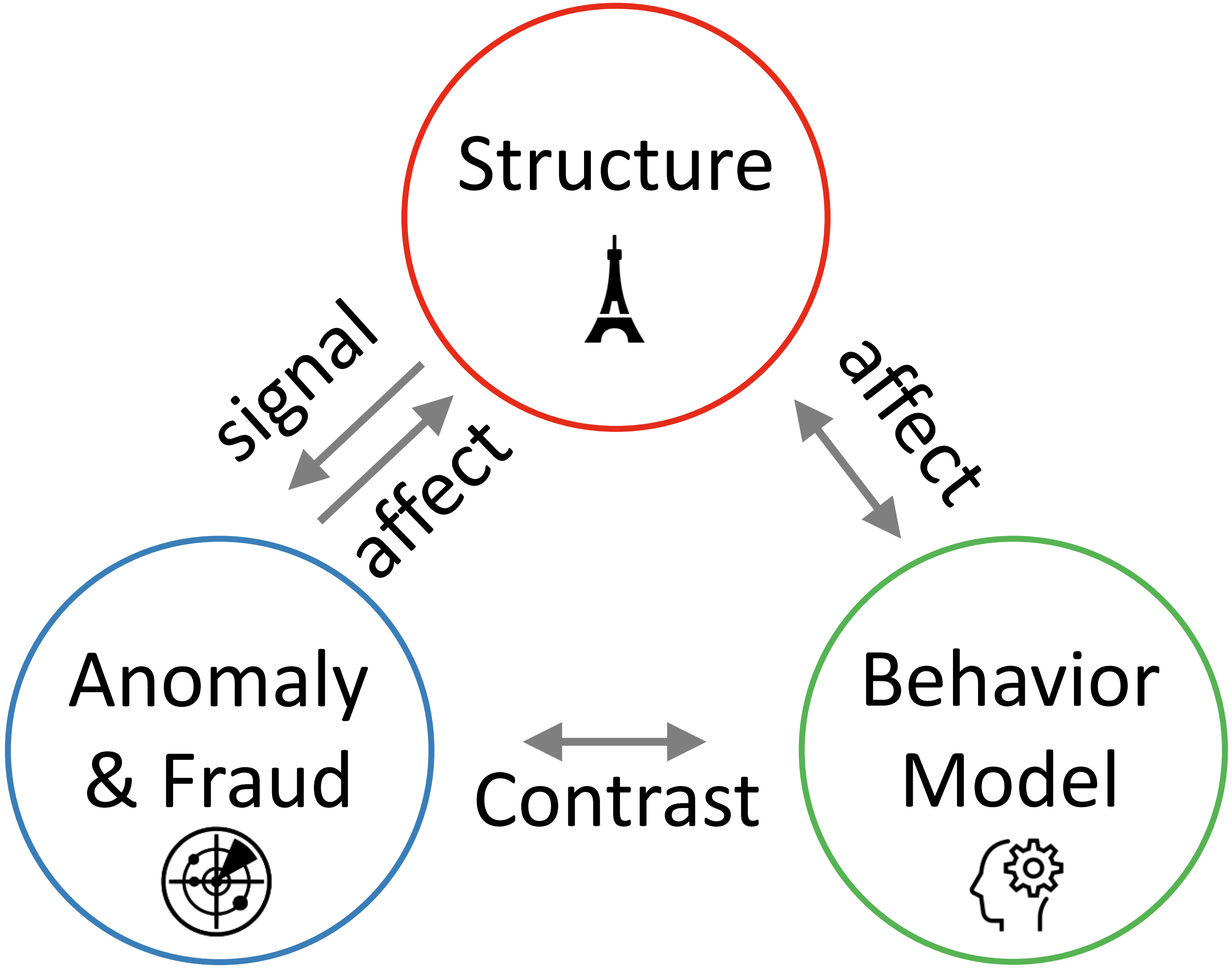
Research Goal and Tasks

- Goal:

> *To Understand*
> *Large Dynamic* ***Graphs*** *and* ***Tensors***
> *on* ***User Behavior***

- Tasks
  - T1. Structure Analysis 
  - T2. Anomaly Detection 
  - T3. Behavior Modeling 

Tasks


Structure
signal
affect
affect
Anomaly
& Fraud
Contrast
Behavior
Model

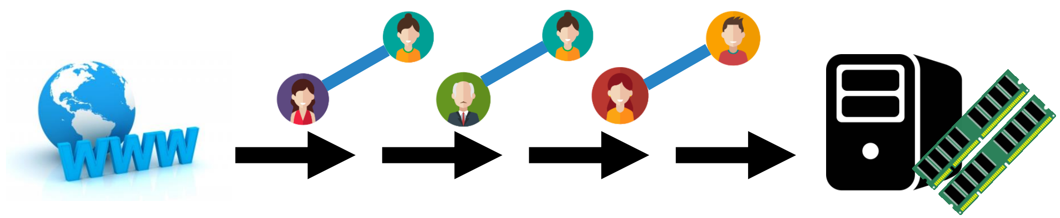
Completed Work by Topics

	T1. Structure Analysis	T2. Anomaly Detection	T3. Behavior Modeling
Graphs	Triangle Count [ICDM17][PAKDD18] [submitted to KDD] Degeneracy [ICDM16]* [KAIS18]*	Anomalous Subgraph [ICDM16]* [KAIS18]*	Purchase Behavior [IJCAI17]
Tensors	Summarization [WSDM17]	Dense Subtensors [PKDD16][WSDM17] [KDD17][TKDD18]	Progressive Behavior [WWW18]

* Duplicated

Approaches (Tools)

- A1. Distributed or external-memory algorithms
- A2. Streaming algorithms based on sampling



- A3. Approximation algorithms
- and their combinations

Roadmap

- Overview
- **Completed Work <<**
  - T1. Structure Analysis
  - T2. Anomaly Detection
  - T3. Behavior Modeling
- Proposed Work
- Conclusion

Completed Work by Topics

	T1. Structure Analysis	T2. Anomaly Detection	T3. Behavior Modeling
Graphs	Triangle Count [ICDM17][PAKDD18] [submitted to KDD] Degeneracy [ICDM16]* [KAIS18]*	Anomalous Subgraph [ICDM16]* [KAIS18]*	Purchase Behavior [IJCAI17]
Tensors	Summarization [WSDM17] skip	Dense Subtensors [PKDD16][WSDM17] [KDD17][TKDD18]	Progressive Behavior [WWW18]

* Duplicated

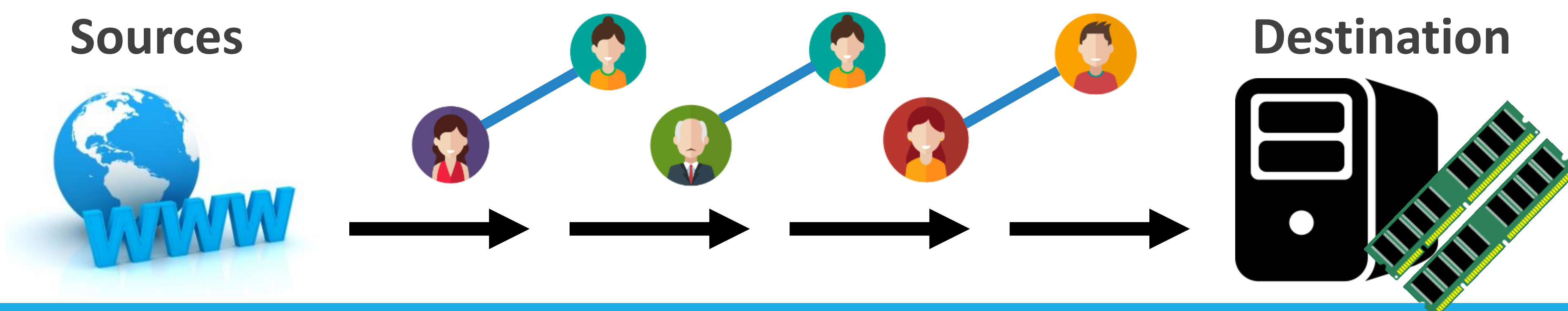
Roadmap

- Overview
- Completed Work
  - T1. Structure Analysis
    - **T1.1 Waiting-Room Sampling <<**
    - T1.2-T1.3 Related Completed Work
  - T2. Anomaly Detection
  - T3. Behavior Modeling
- Proposed Work
- Conclusion

Kijung Shin, “WRS: Waiting Room Sampling for Accurate Triangle Counting in Real Graph Streams”, ICDM 2017

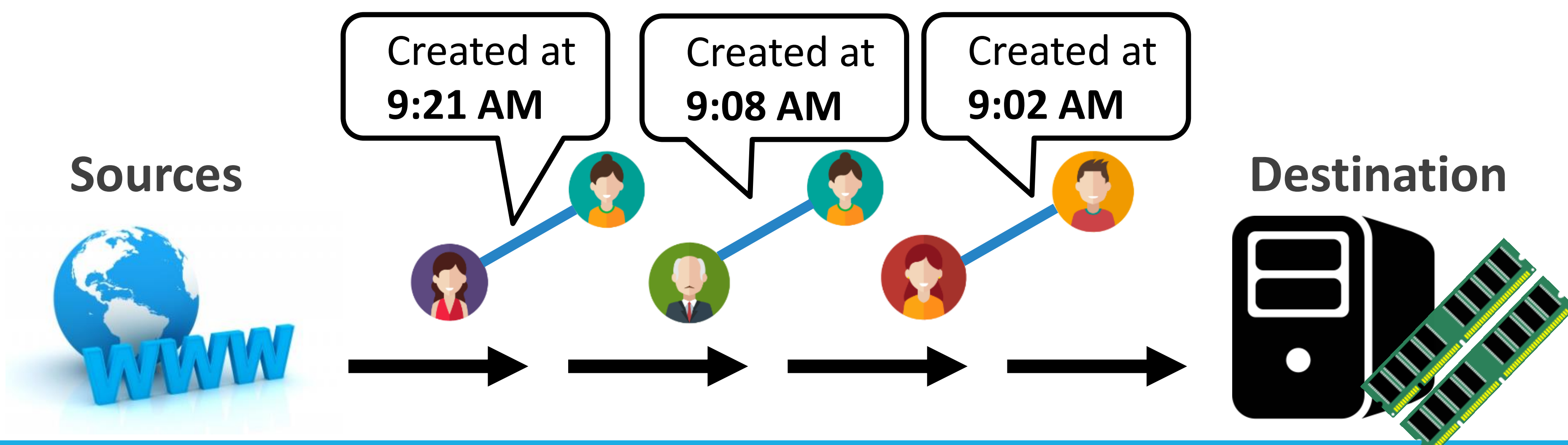
Graph Stream Model

- Widely-used data model for graphs
- **Sequence of edges**
  - graph is given over time as a sequence of edges
  - appropriate for **dynamic graphs**
- **Limited memory**
  - cannot store all edges in the stream
  - only samples or summaries
  - appropriate for **large graphs**



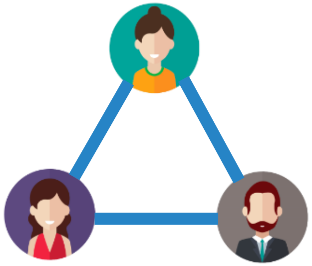
Relaxed Graph Stream Model

- **Chronological order**
  - edges are streamed in the order that they are created
  - natural for **dynamic graphs**
  - **temporal patterns can** exist
  - algorithms can **exploit** the patterns

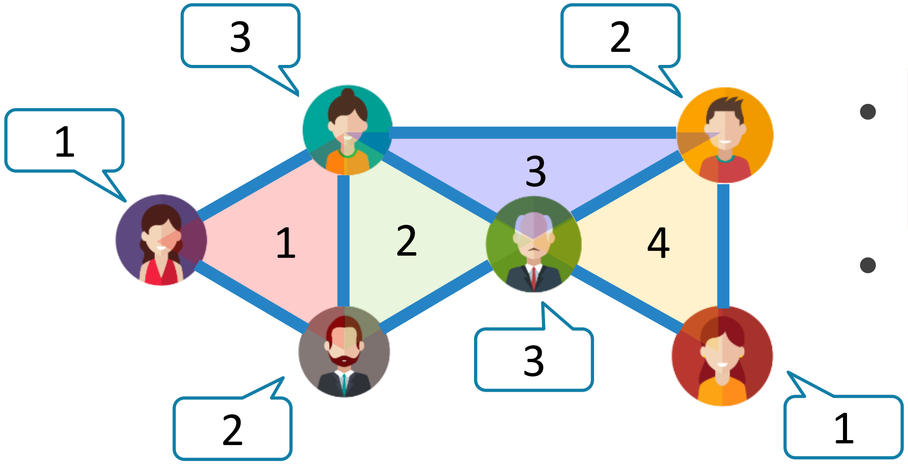


Triangles in a Graph

- **A triangle** is 3 nodes connected to each other



- **The count of triangles** has many applications
  - Community detection, spam detection, query optimization



- **Global triangle count**: count of all triangles in the graph
- **Local triangle count**: count of the triangles incident to each node

Problem Definition

- **Given**:
  - a sequence of edges in the **chronological order**
  - memory budget k (i.e., up to k edges can be stored)
- **Estimate:** count of global triangles
- **To Minimize:** estimation error

"What are temporal patterns in real graph streams?"

"How can we exploit the patterns for accurate triangle counting?"

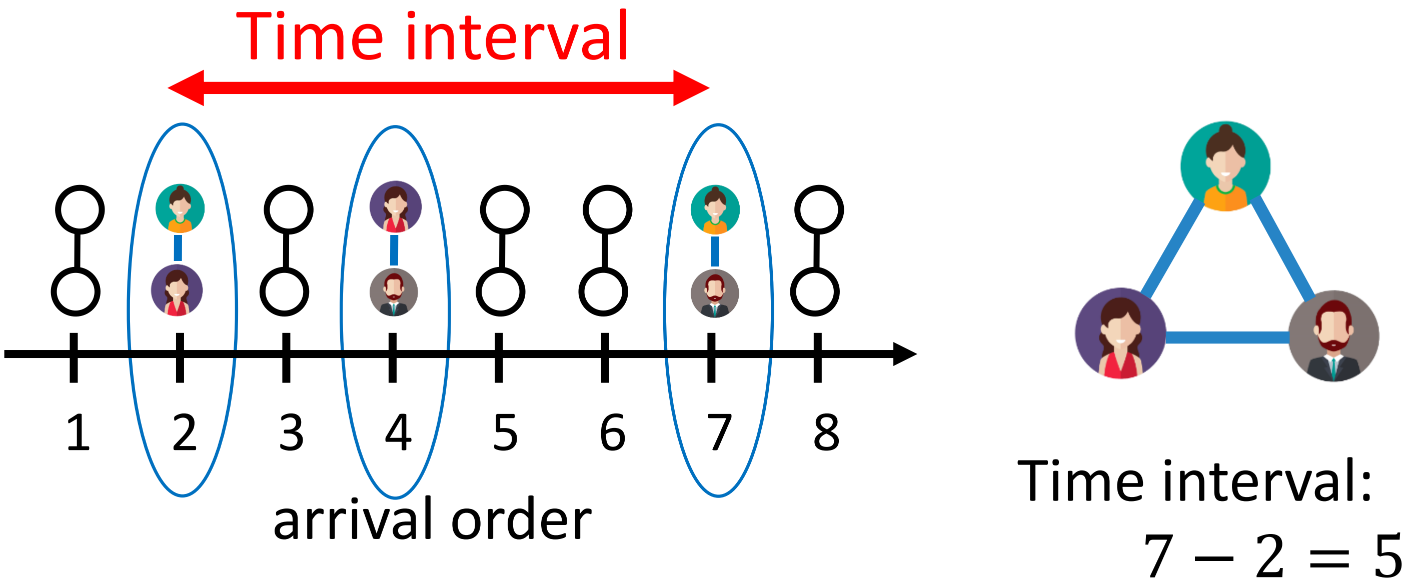
Roadmap

- Overview
- Completed Work
  - T1. Structure Analysis
    - T1.1 Waiting-Room Sampling
      - **Temporal Pattern <<**
      - Algorithm
      - Experiments
    - T1.2-T1.3 Related Completed Work
  - T2. Anomaly Detection
  - T3. Behavior Modeling
- Proposed Work
- Conclusion

Time Interval of a Triangle

- **Time interval** of a triangle:

arrival order of its last edge	–	arrival order of its first edge

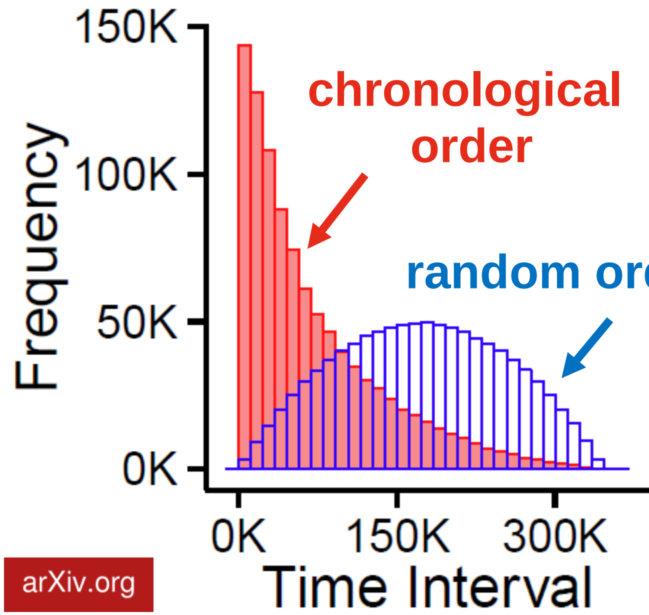


Detail

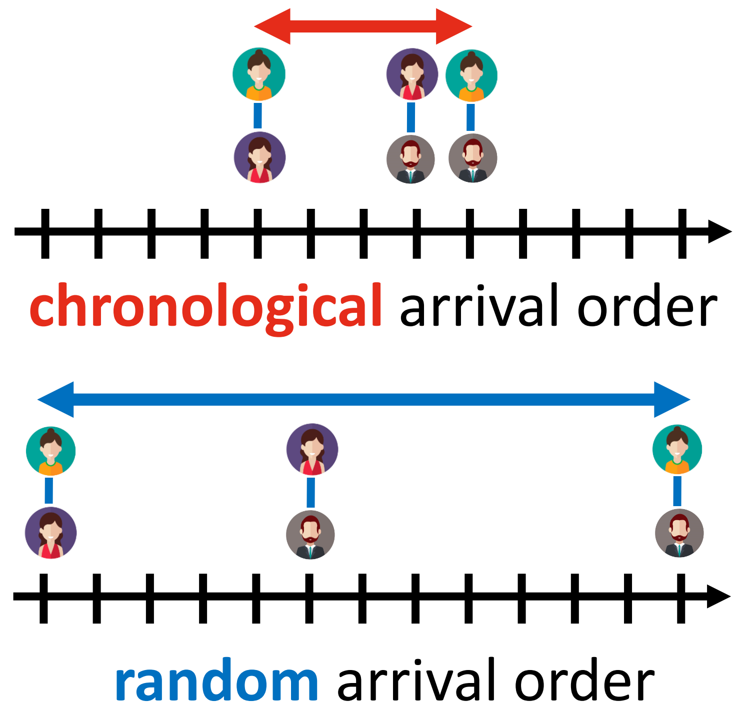
Time Interval Distribution

- **Temporal Locality:**
  - average time interval is
  - **2X shorter** in the **chronological order**
  - than in a **random order**

arXiv.org

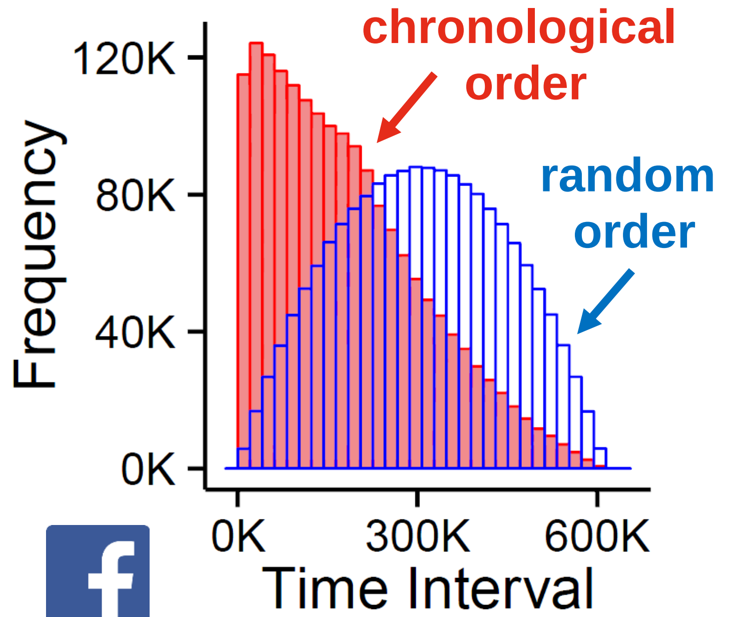


arXiv.org



Temporal Locality

- One interpretation:
  - edges are more likely to form
  - triangles with **edges close in time**
  - than with **edges far in time**
- Another interpretation:
  - **new edges** are more likely to form
  - triangles with **recent edges**
  - than with **old edges**



"How can we exploit temporal locality for accurate triangle counting?"

Roadmap

- Overview
- Completed Work
  - T1. Structure Analysis
    - T1.1 Waiting-Room Sampling
      - Temporal Pattern
      - **Algorithm <<**
      - Experiments
    - T1.2-T1.3 Related Completed Work
  - T2. Anomaly Detection
  - T3. Behavior Modeling
- Proposed Work
- Conclusion

Algorithm Overview

- Δ: estimate of triangle count
- p_{uvw}: probability that triangle (u, v, w) is discovered

(1) Arrival Step

new edge $u - v$

$u - x$	$u - y$	$v - x$	$v - y$

memory

(2) Counting Step

$u - v$

$u - v$, $u - y$, $v - y$

$u - x$	$u - y$	$v - x$	$v - y$

$\Delta \leftarrow \Delta + 1/p_{uvy}$

(3) Sampling Step

$u - x$	$u - v$	$v - x$	$v - y$

Algorithm Overview (cont.)

- Δ: estimate of triangle count
- p_{uvw}: probability that triangle (u, v, w) is discovered

(1) Arrival Step

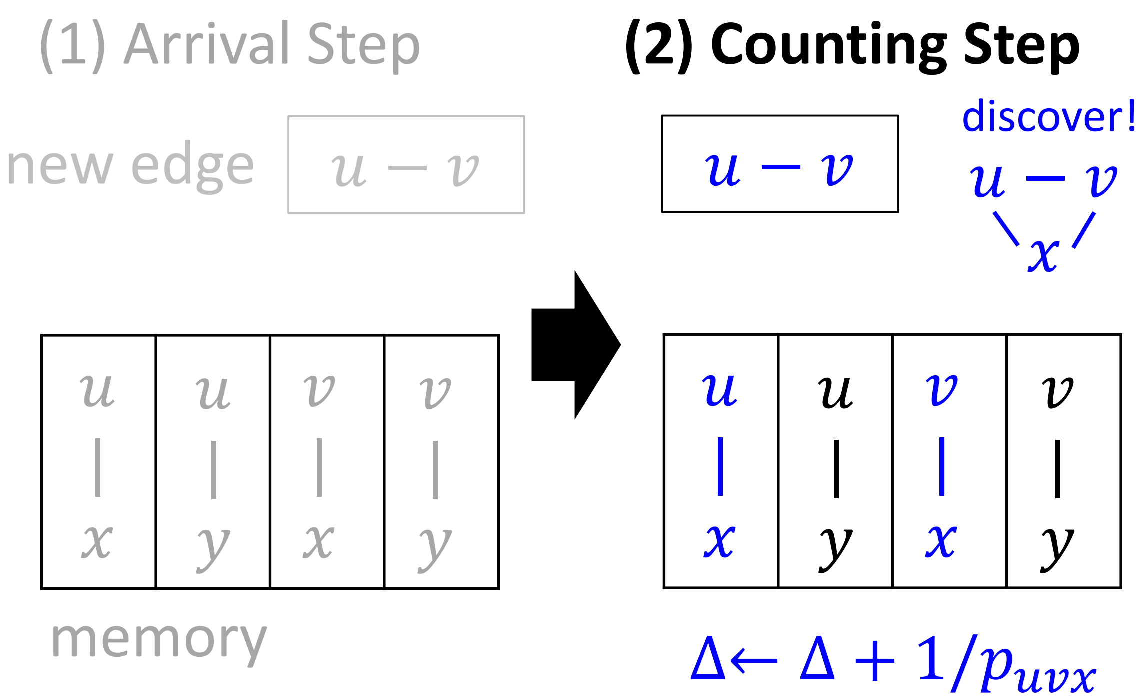
new edge $\boxed{u - v}$

u	u	v	v
$\vert$	$\vert$	$\vert$	$\vert$
x	y	x	y

memory

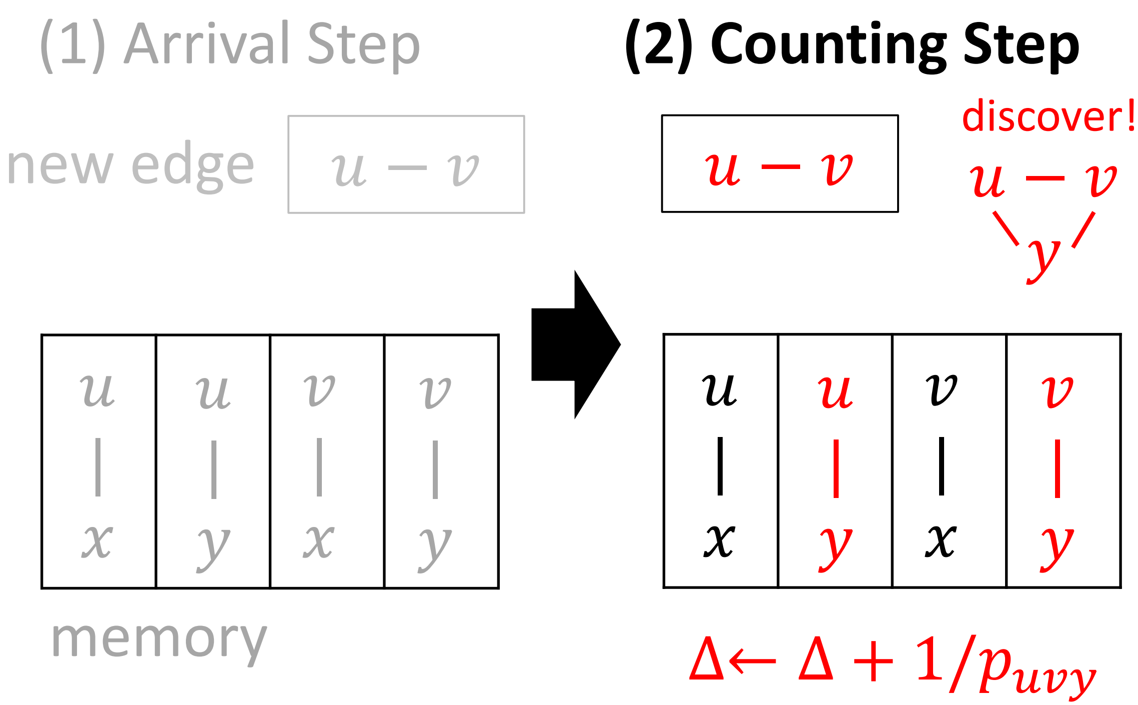
Algorithm Overview (cont.)

- Δ: estimate of triangle count
- p_{uvw}: probability that triangle (u, v, w) is discovered



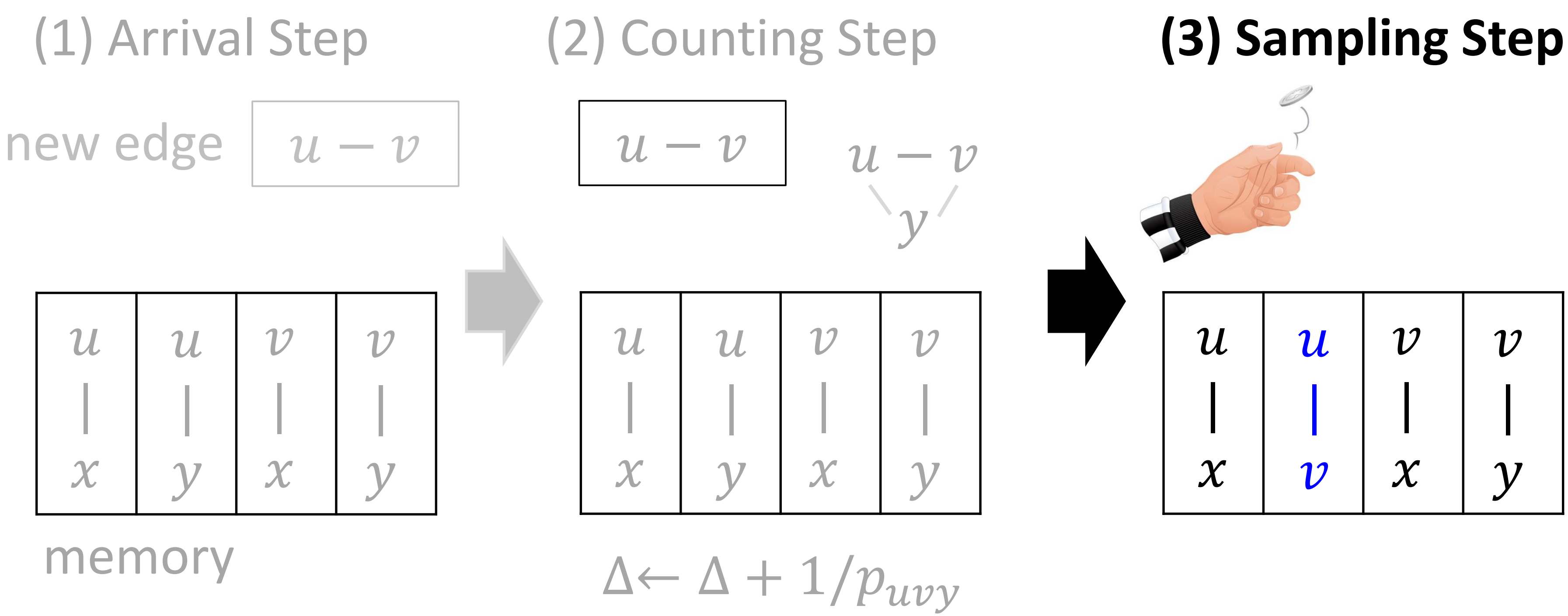
Algorithm Overview (cont.)

- Δ: estimate of triangle count
- p_{uvw}: probability that triangle (u, v, w) is discovered



Algorithm Overview (cont.)

- Δ: estimate of triangle count
- p_{uvw}: probability that triangle (u, v, w) is discovered



Detail

Goal of Sampling Step

- to maximize **discovering probability** p_{uvw}

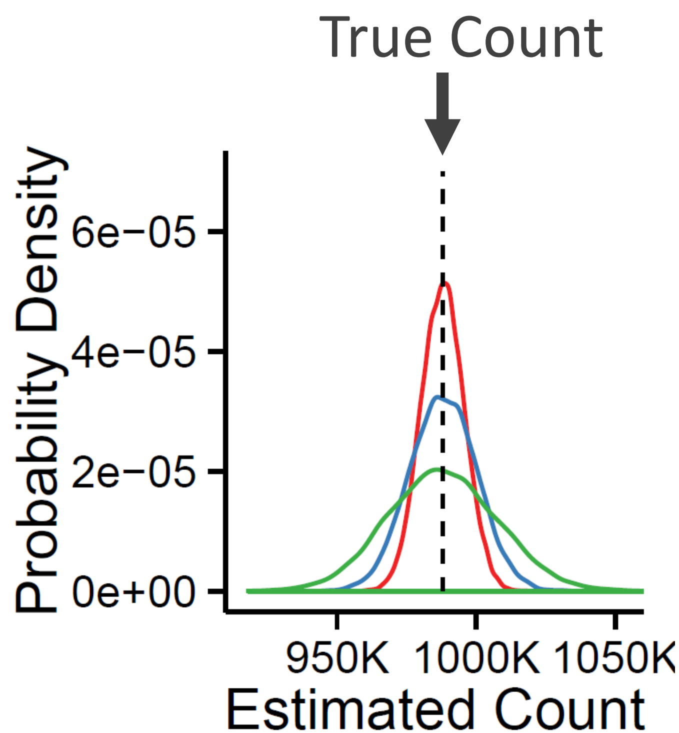
Theorem. **Variance** of our estimate:

$$\mathrm{Var}[\Delta] \approx \sum_{(u,v,w)} (1/p_{uvw} - 1)$$

Theorem. **Unbiasedness** of our estimate:

$$\mathrm{Bias}[\Delta] = \mathrm{Exp}[\Delta] - \text{True count} = 0$$

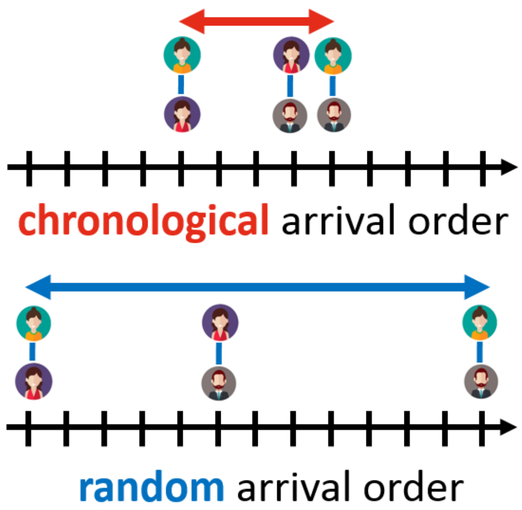
$$Estimation\ Error\ = \underbrace{\cancel{Bias}}_{0} + Variance$$



Increasing Discovering Prob.

> *"How can we increase discovering probabilities of triangles?"*

- Recall Temporal Locality:
  - new edges are more likely to form
  - triangles with **recent edges**
  - than with **old edges**
- **Waiting-Room Sampling** (WRS)
  - treats **recent edges better** than old edges
  - to exploit temporal locality



Waiting-Room Sampling (WRS)

- Divides memory space into two parts

Waiting Room: latest edges are ***always* stored**

Reservoir: the remaining edges are **sampled**

New edge $\boxed{e_{80}}$

Waiting Room (FIFO)

e_{79}	e_{78}	e_{77}	e_{76}

α% of budget

Reservoir (Random Replace)

e_{61}	e_{7}	e_{18}	e_{25}	e_{40}	e_{1}	e_{28}

$(100-\alpha)$% of budget

WRS: Sampling Steps (Step 1)

New edge $\boldsymbol{e_{80}}$

Waiting Room (FIFO)

e_{79}	e_{78}	e_{77}	$\boldsymbol{e_{76}}$

Reservoir (Random Replace)

e_{61}	e_{7}	e_{18}	e_{25}	e_{40}	e_{1}	e_{28}

Popped edge $\boldsymbol{e_{76}}$

Waiting Room (FIFO)

$\boldsymbol{e_{80}}$	e_{79}	e_{78}	e_{77}

Reservoir (Random Replace)

e_{61}	e_{7}	e_{18}	e_{25}	e_{40}	e_{1}	e_{28}

WRS: Sampling Steps (Step 2)

Popped edge $\boxed{e_{76}}$

Waiting Room (FIFO)

e_{80}	e_{79}	e_{78}	e_{77}

Reservoir (Random Replace)

e_{61}	e_7	e_{18}	e_{25}	e_{40}	e_1	e_{28}

store

replace!

e_{61}	e_7	e_{18}	e_{25}	e_{76}	e_1	e_{28}

or

or

discard

e_{61}	e_7	e_{18}	e_{25}	e_{40}	e_1	e_{28}

Summary of Algorithm

Waiting-Room Sampling!

(1) Arrival Step

new edge $u - v$

$u - x$	$u - y$	$v - x$	$v - y$

memory

(2) Discovery Step

$u - v$

discover!

$u - v$, $u \setminus x$, v / x

$u - x$	$u - y$	$v - x$	$v - y$

$$\Delta \leftarrow \Delta + 1/p_{uvx}$$

(3) Sampling Step

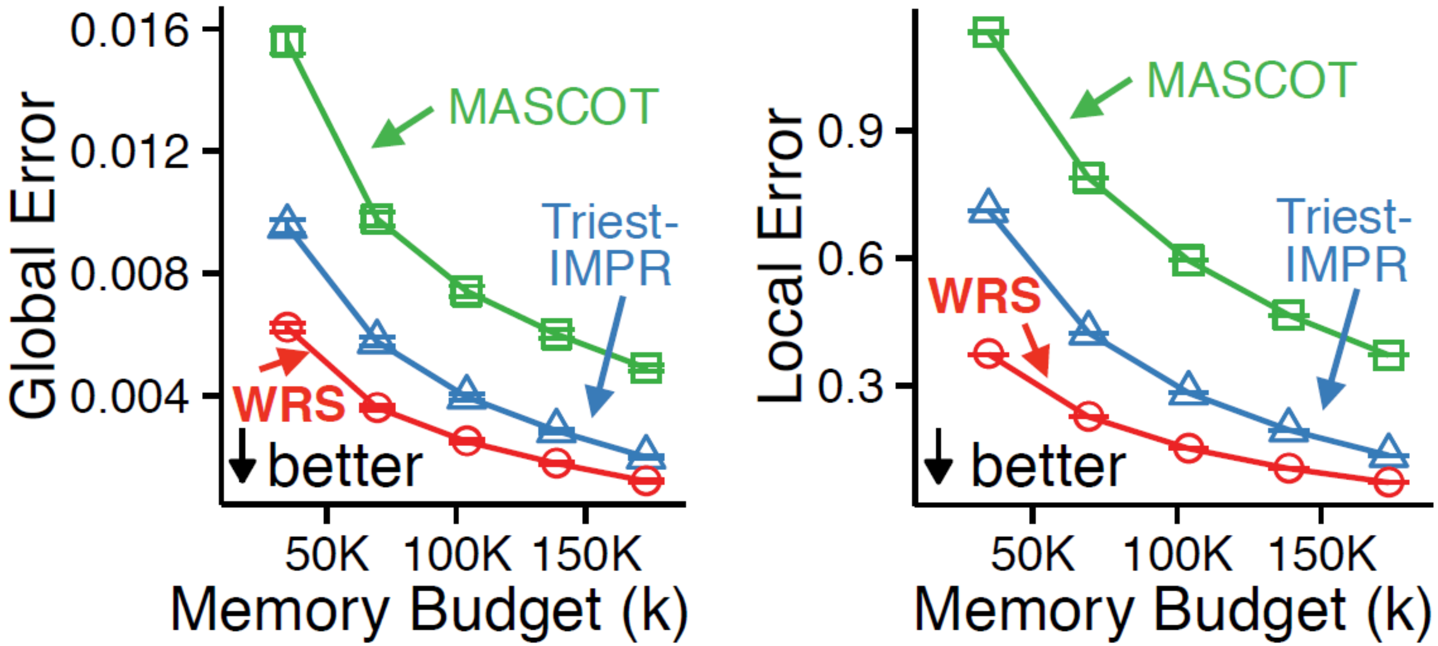
$u - x$	$u - v$	$v - x$	$v - y$

Roadmap

- Overview
- Completed Work
  - T1. Structure Analysis
    - T1.1 Waiting-Room Sampling
      - Temporal Pattern
      - Algorithm
      - **Experiments <<**
    - T1.2-T1.3 Related Completed Work
  - T2. Anomaly Detection
  - T3. Behavior Modeling
- Proposed Work
- Conclusion

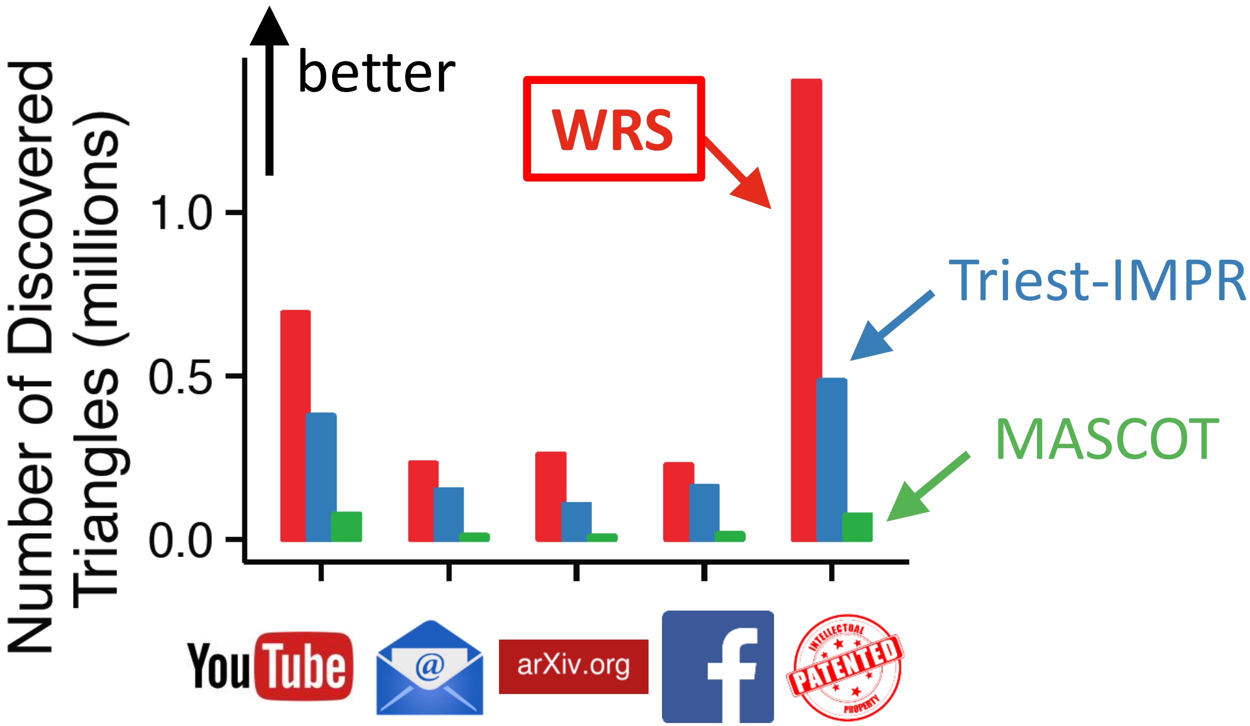
Experimental Results: Accuracy

- Datasets:
- WRS is most accurate (reduces error up to $\mathbf{47\%}$)



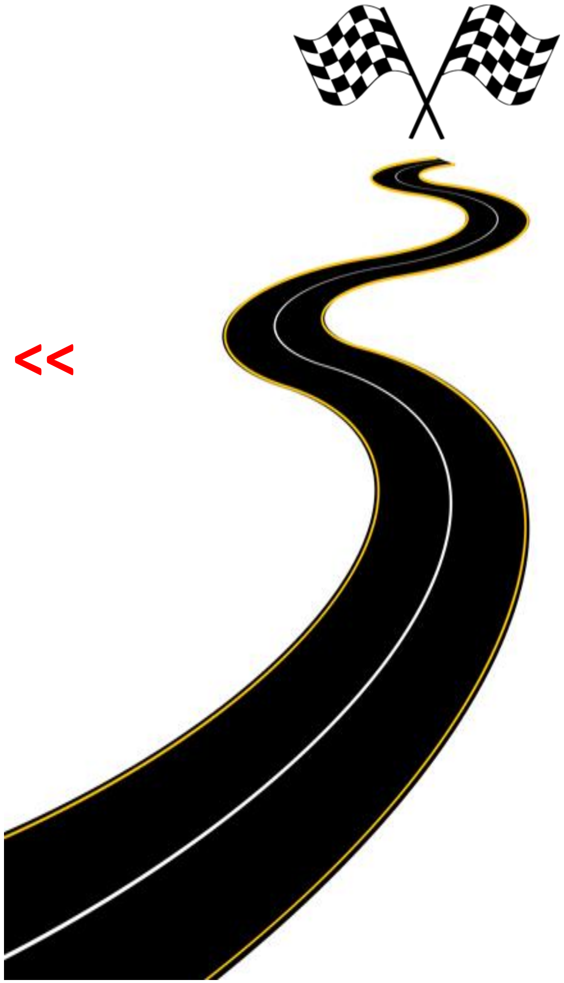
Discovering Probability

- WRS increases discovering probability p_{uvw}
- WRS discovers up to $3 \times$ more triangles



Roadmap

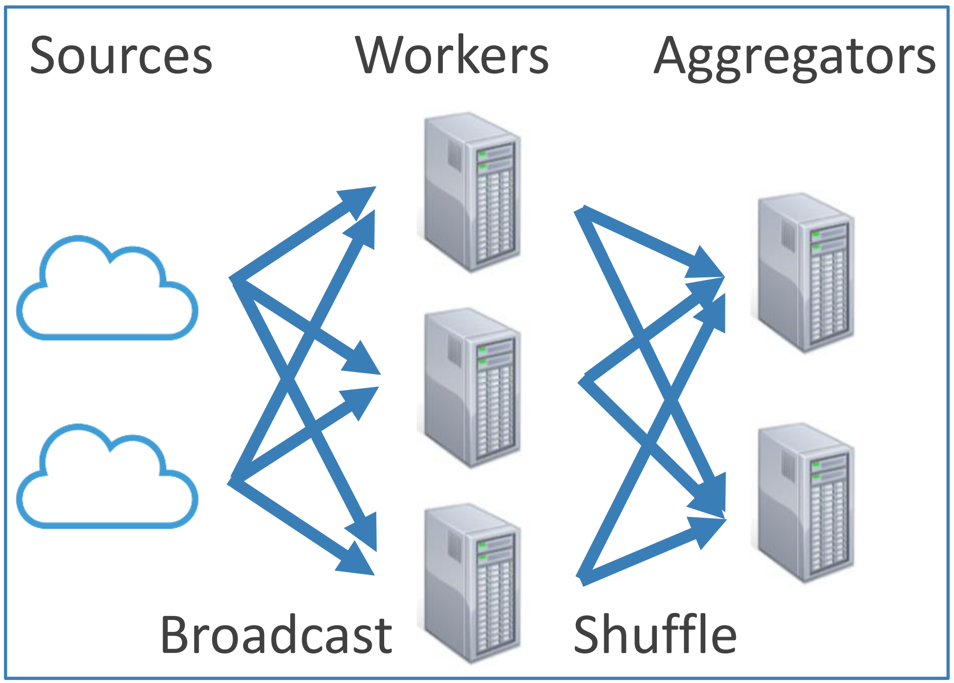
- Overview
- Completed Work
  - T1. Structure Analysis 
    - T1.1 Waiting-Room Sampling
    - **T1.2-T1.3 Related Completed Work <<**
  - T2. Anomaly Detection 
  - T3. Behavior Modeling 
- Proposed Work
- Conclusion



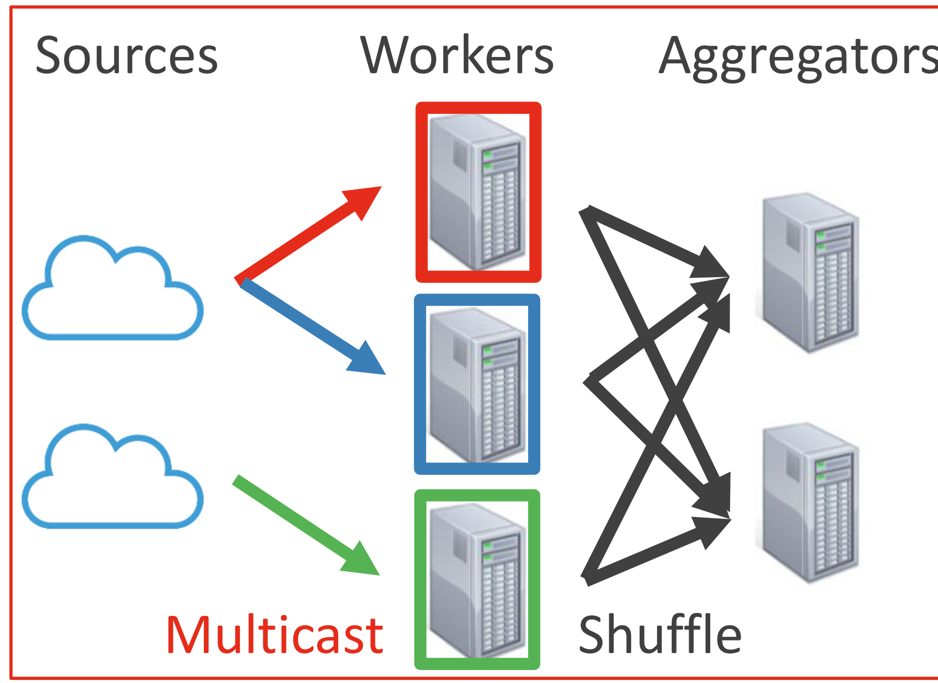
T1.2 Distributed Counting of Triangles

- Goal: to utilize ***multiple machines*** for triangle counting in a graph stream?

Tri-Fly [PAKDD18]



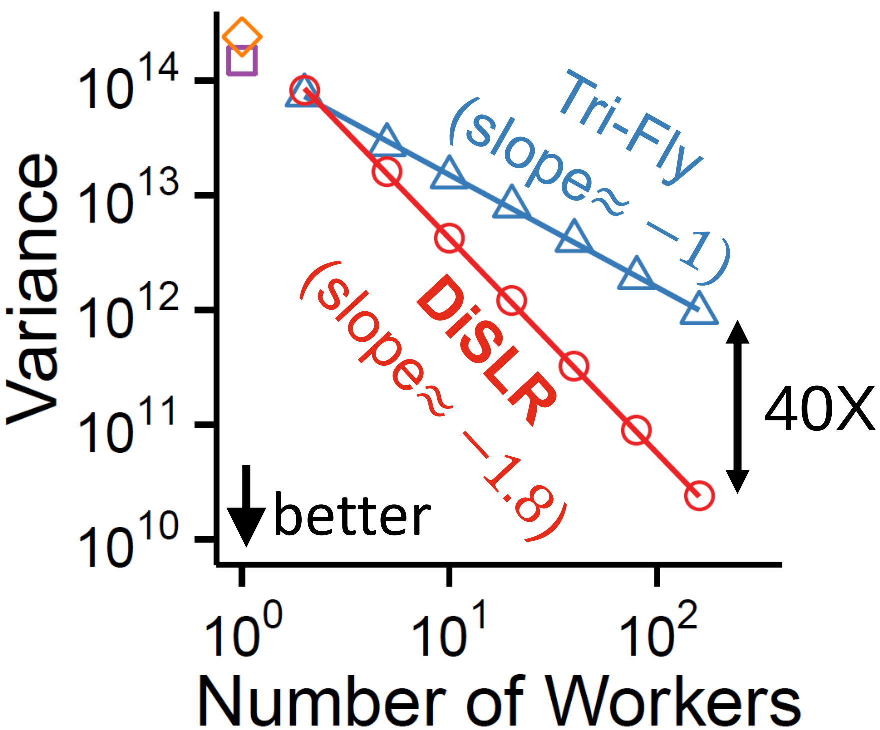
DiSLR [submitted to KDD]

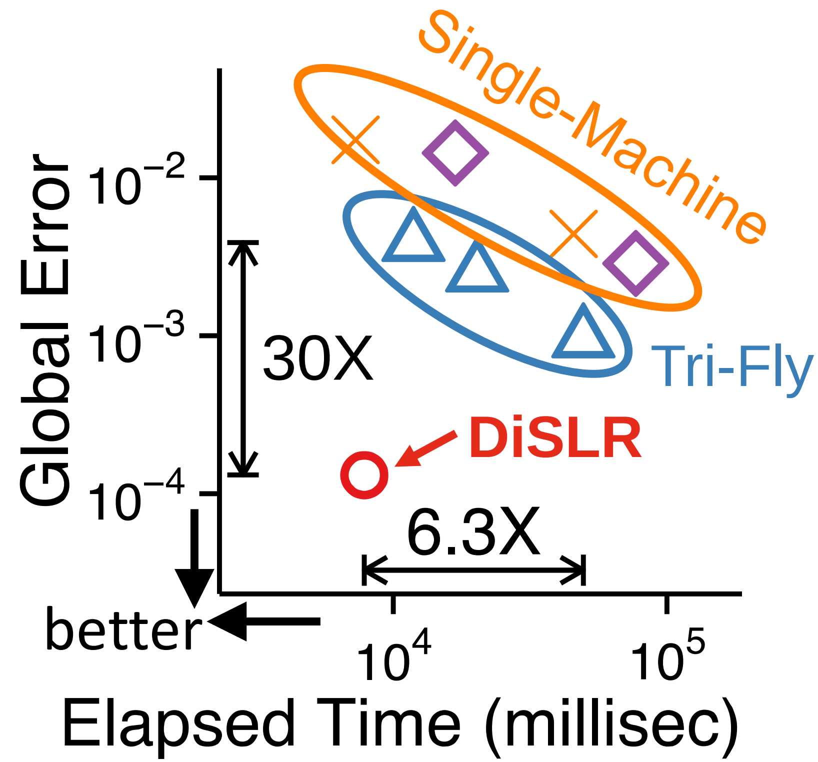


Kijung Shin, Mohammad Hammoud, Euiwoong Lee, Jinoh Oh, and Christos Faloutsos, "Tri-Fly: Distributed Estimation of Global and Local Triangle Counts in Graph Streams", PAKDD 2018

T1.2 Performance of Tri-Fly and DiSLR

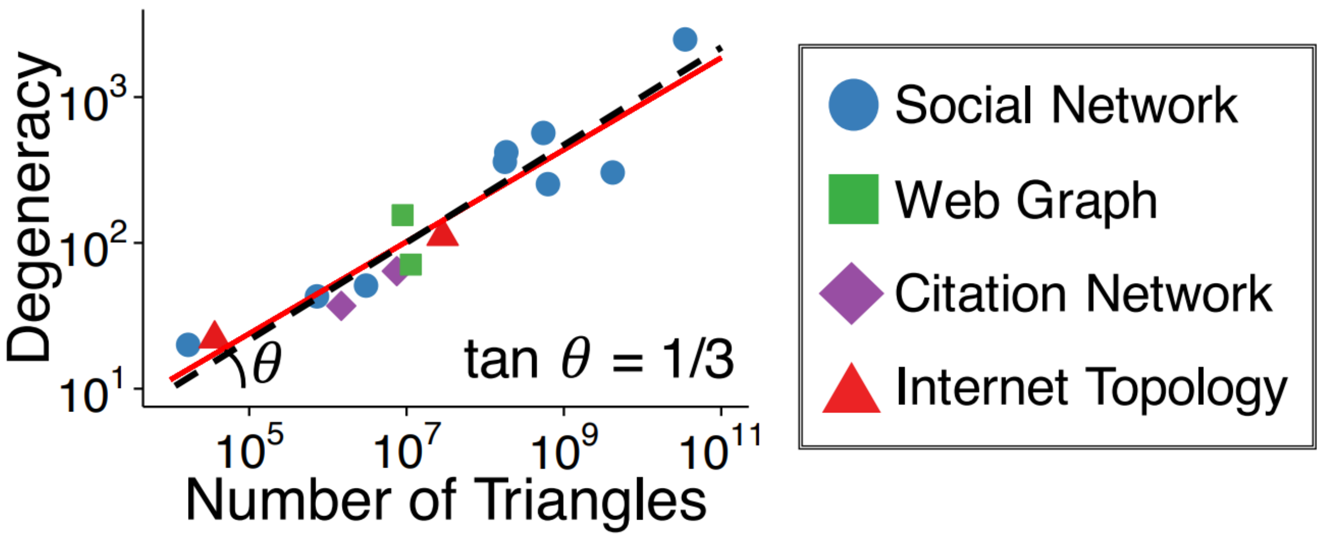
- $Estimation\ Error\ = \underset{0}{\cancel{Bias}} + Variance$





T1.3 Estimation of Degeneracy

- Goal: to estimate the ***degeneracy**** in a graph stream?
- ***Core-Triangle Pattern***
  - 3:1 power law between the triangle count and the degeneracy



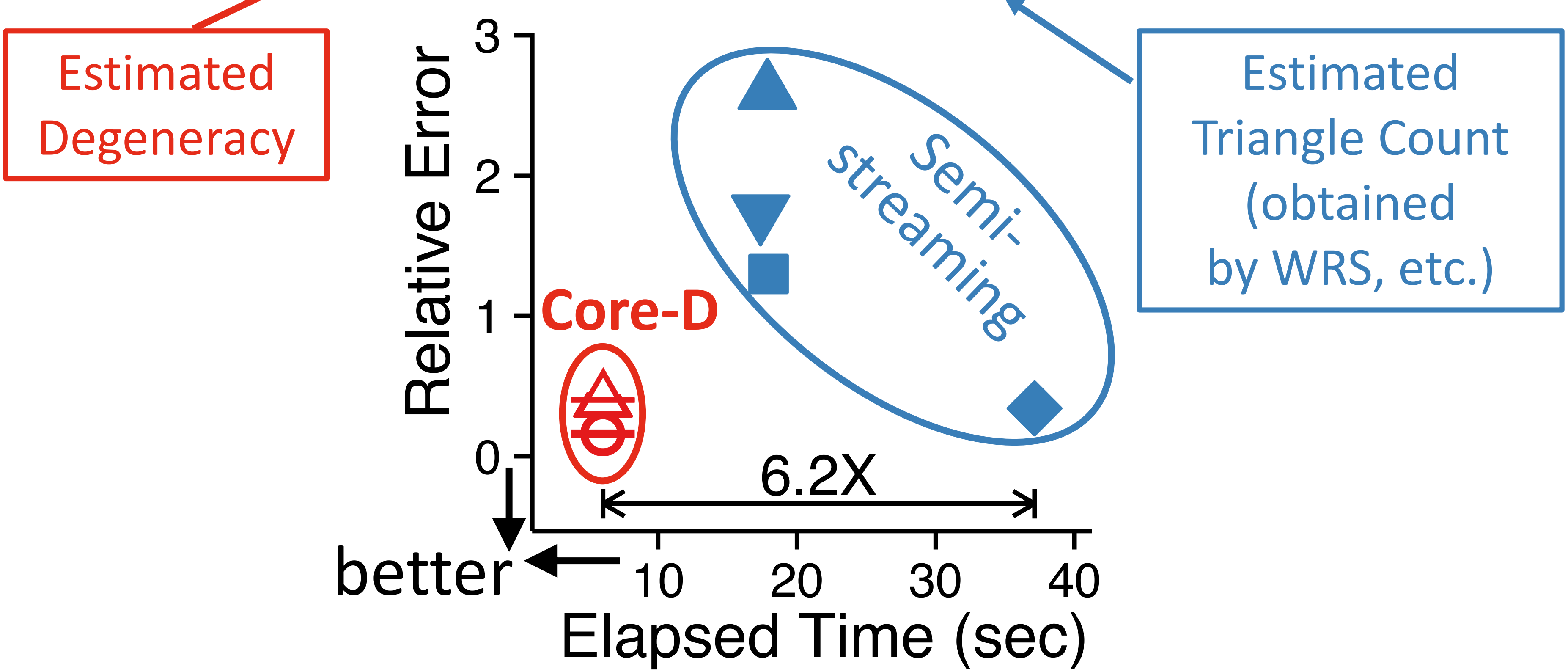
*degeneracy: maximum k such that a subgraph where every node has degree at least k exists.

Kijung Shin, Tina Eliassi-Rad, and Christos Faloutsos, "Patterns and Anomalies in kCores of Real-world Graphs with Applications", KAIS 2018 (previously ICDM 2016)

T1.3 Core-D Algorithm

- **Core-D**: one-pass streaming algorithm for degeneracy

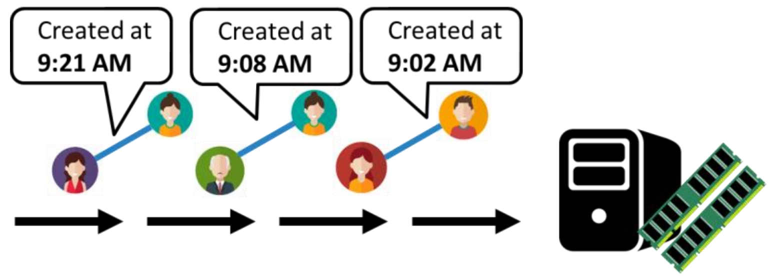
$$\hat{d} = \exp(\alpha \cdot \log(\widehat{\Delta}) + \beta)$$



Structure Analysis of Graphs

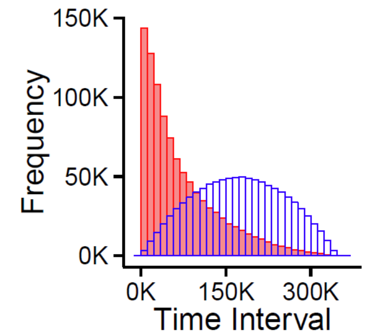
Models:

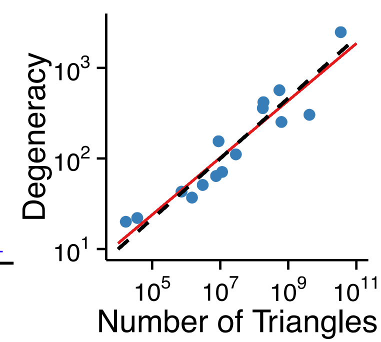
- Relaxed graph stream model
- Distributed graph stream model



Patterns:

- Temporal locality
- Core-Triangle pattern

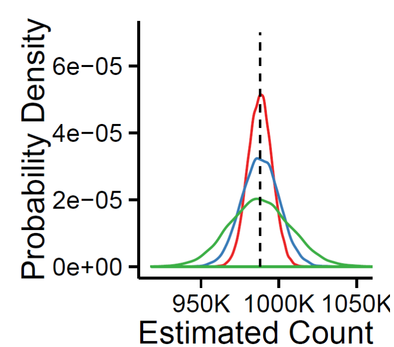




Algorithms:

- WRS, Tri-Fly, and DiSLR
- Core-D

Analyses: bias and variance



Completed Work by Topics

	T1. Structure Analysis	T2. Anomaly Detection	T3. Behavior Modeling
Graphs	Triangle Co [ICDM17][PAKD [submitted to KDD] Degenera [ICDM16]* [KAIS	skip Anomalous Subgraph [ICDM16]* [KAIS18]*	Purchase Behavior [IJCAI17]
Tensors	Summariza skip [WSDM17]	Dense Subtensors [PKDD16][WSDM17] [KDD17][TKDD18]	Progressive Behavior [WWW18]

* Duplicated

Roadmap

- Overview
- Completed Work
  - T1. Structure Analysis
  - T2. Anomaly Detection
    - **T2.1 M-Zoom <<**
    - T2.2-T2.3 Related Completed Work
  - T3. Behavior Modeling
- Proposed Work
- Conclusion

Kijung Shin, Bryan Hooi, and Christos Faloutsos, “Fast, Accurate and Flexible Algorithms for Dense Subtensor Mining”, TKDD 2018 (previously ECML/PKDD 2016)

Motivation: Review Fraud

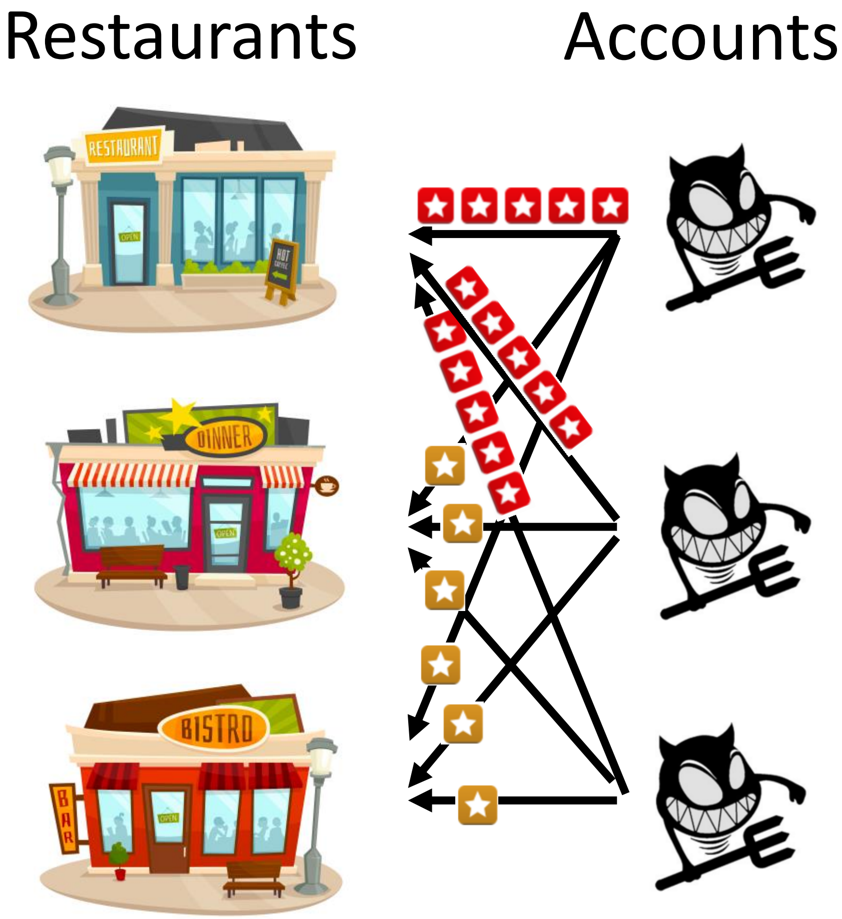
Alice's

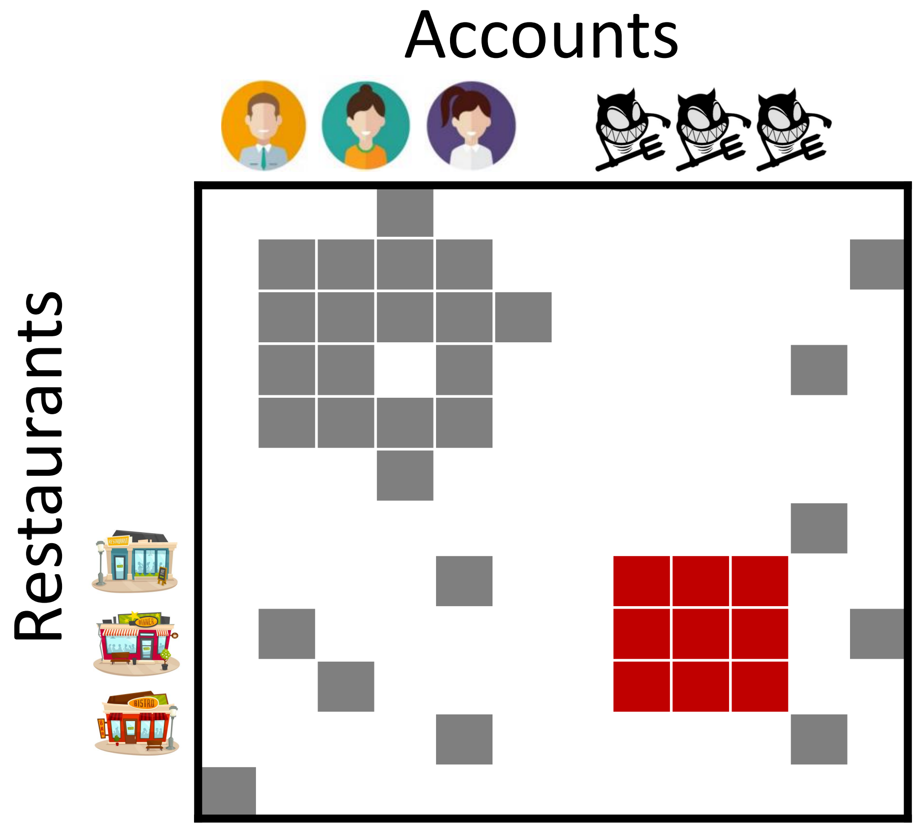
Bob's

Carol's

Alice

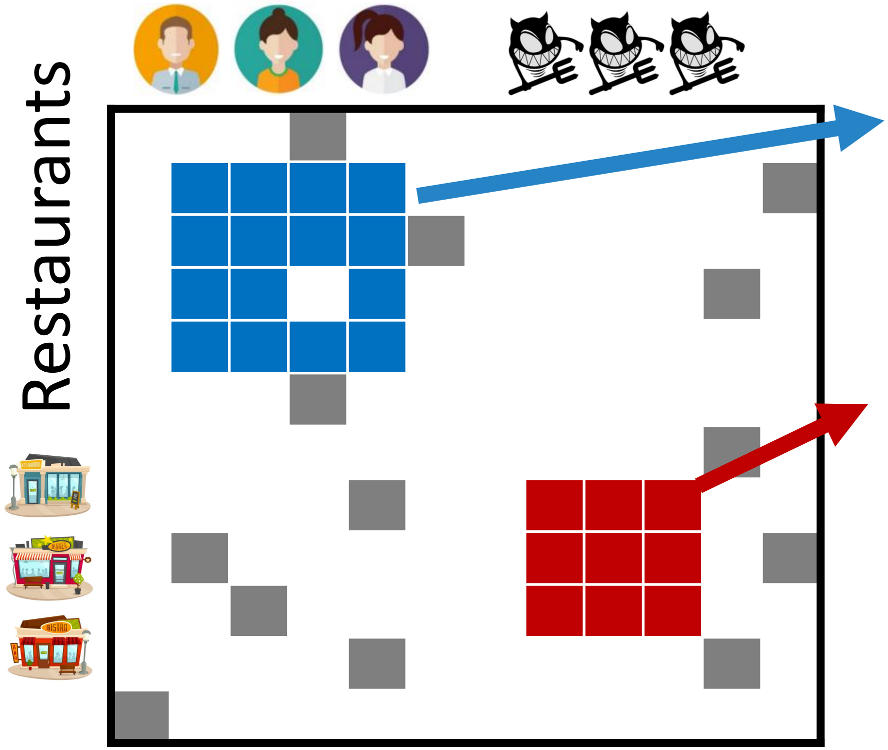
Fraud Forms Dense Block





Adjacency Matrix

Problem: Natural Dense Subgraphs



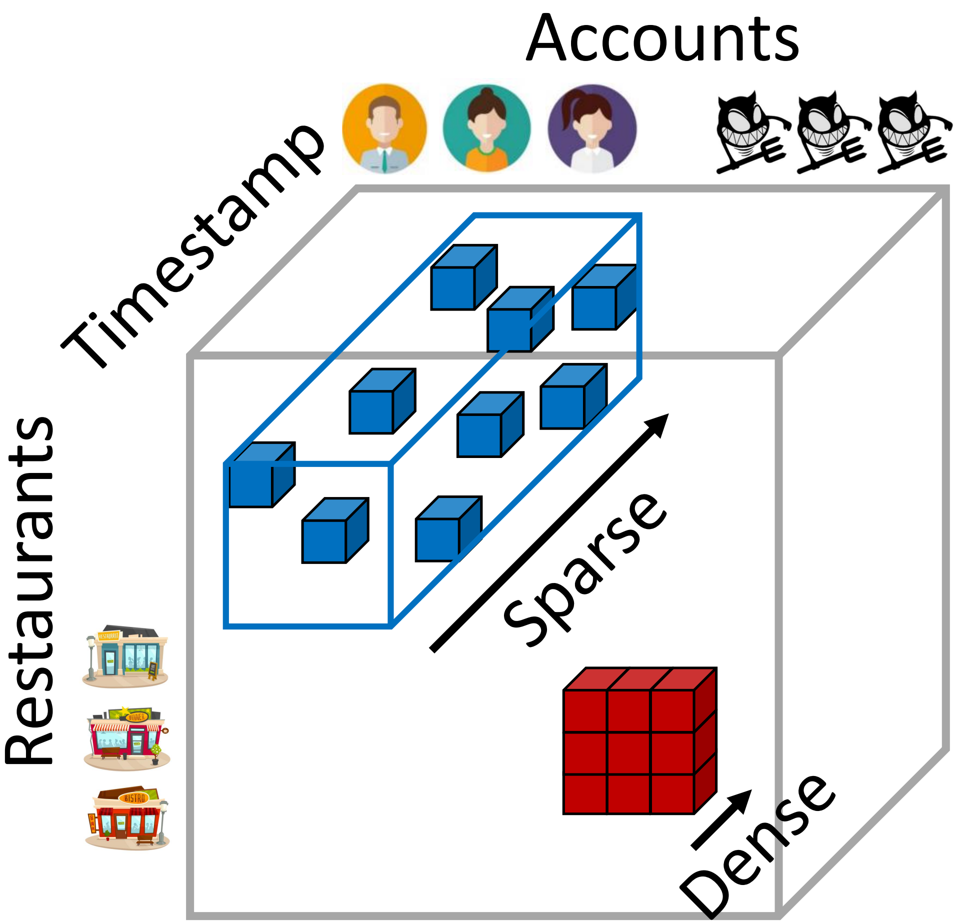
Adjacency Matrix

natural dense blocks
(core, community, etc.)

suspicious dense blocks
formed by fraudsters

- **Question.** How can we distinguish them?

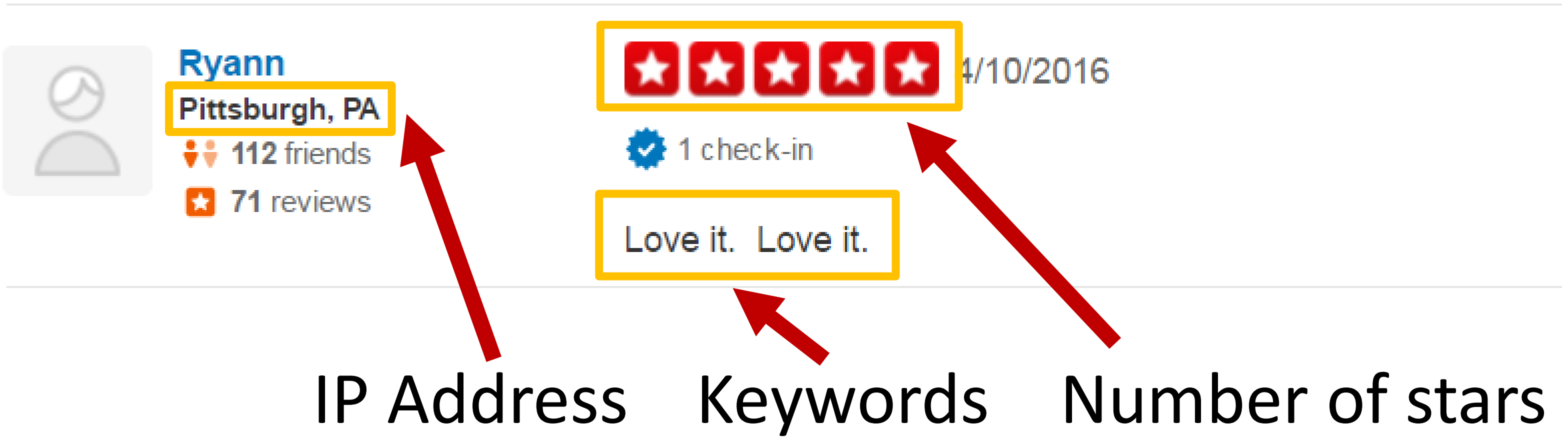
Solution: Tensor Modeling



- Along the time axis…
  - Natural dense blocks are **sparse** (formed gradually)
  - Suspicious dense blocks are **dense** (synchronized behavior)
- In the tensor model
  - Suspicious dense blocks become **denser** than natural dense blocks

Solution: Tensor Modeling (cont.)

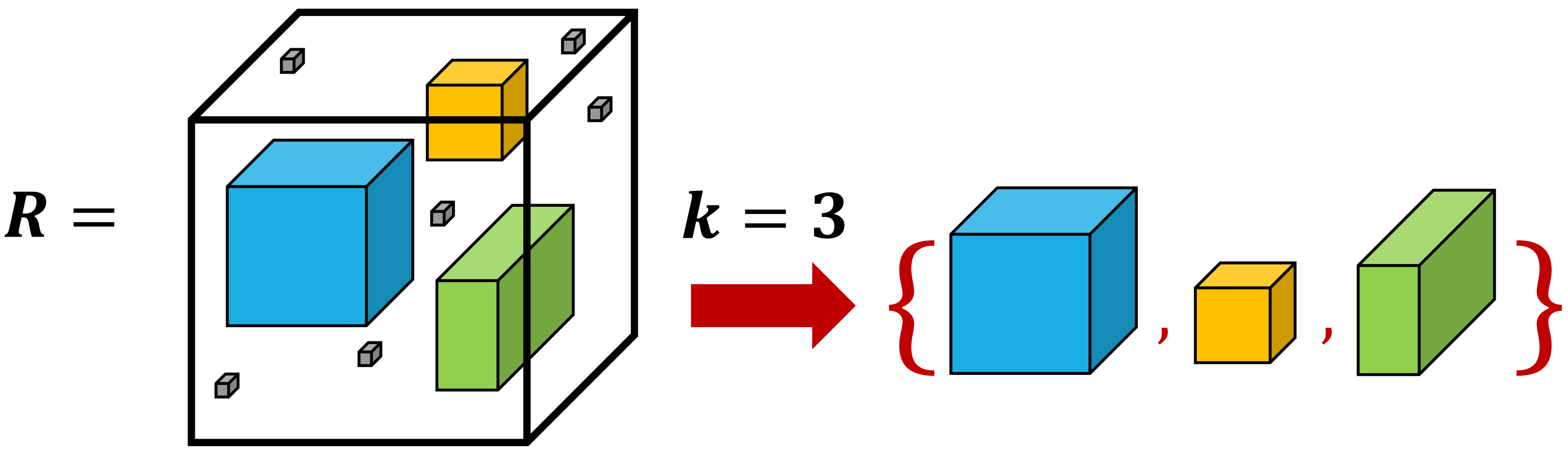
- High-order tensor modeling:
  - any side information can be used additionally



> *"Given a large-scale high-order tensor,*
> *how can we find dense blocks in it?"*

Problem Definition

- **Given**: (1) $\boldsymbol{R}$: an N-order tensor,
  (2) $\boldsymbol{\rho}$: a density measure,
  (3) $\boldsymbol{k}$: the number of blocks we aim to find
- **Find**: $\boldsymbol{k}$ distinct dense blocks maximizing $\boldsymbol{\rho}$



Density Measures

- How should we define "density" (i.e., ρ)?
  - no one absolute answer
  - depends on data, types of anomalies, etc.
- Goal: flexible algorithm working well with various reasonable measures

✓ Arithmetic avg. degree ρ_A

✓ Geometric avg. degree ρ_G

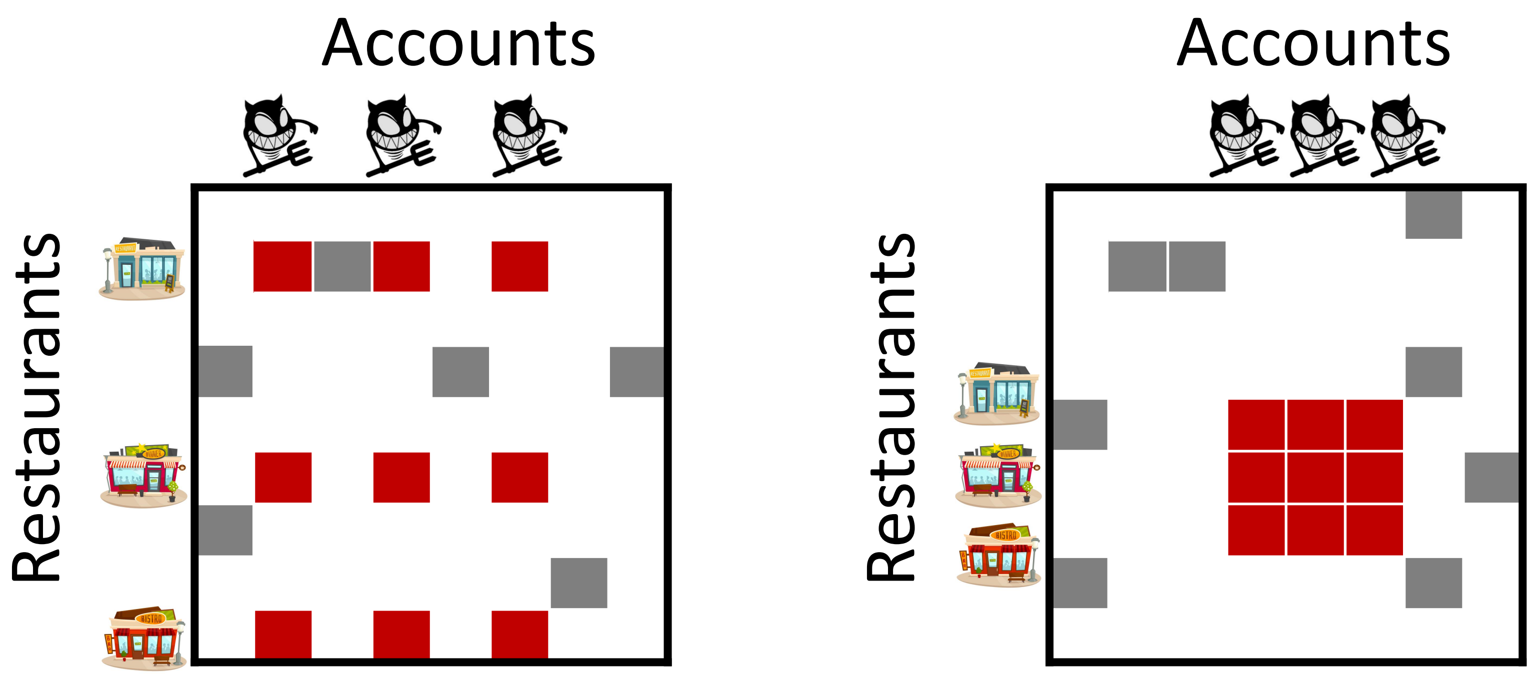
✓ Suspiciousness (KL Divergence) ρ_S

⦸ Traditional Density: $\rho_T(B) = \mathrm{EntrySum}(B)/\mathrm{Vol(B)}$

- maximized by a single entry with the maximum value

Clarification of Blocks (Subtensors)

- The concept of blocks (subtensors) is independent of the orders of rows and columns
- Entries in a block do not need to be adjacent

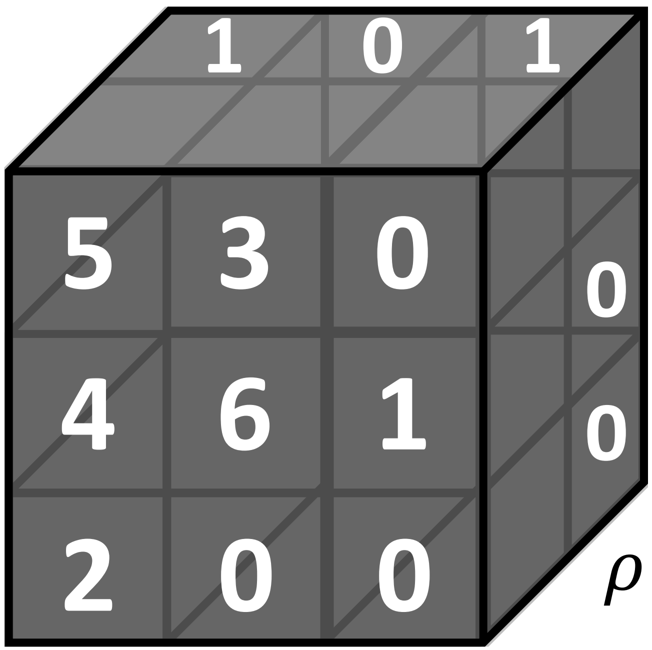


Roadmap

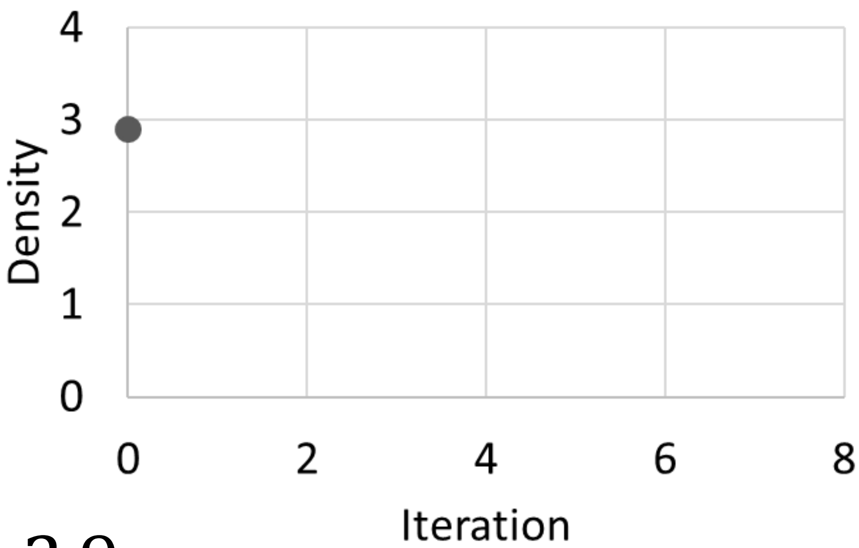
- Overview
- Completed Work
  - T1. Structure Analysis
  - T2. Anomaly Detection
    - T2.1 M-Zoom [PKDD 16]
      - **Algorithm <<**
      - Experiments
    - T2.2-T2.3 Related Completed Work
  - T3. Behavior Modeling
- Proposed Work
- Conclusion

Single Dense Block Detection

- **Greedy search**
- Starts from the entire tensor

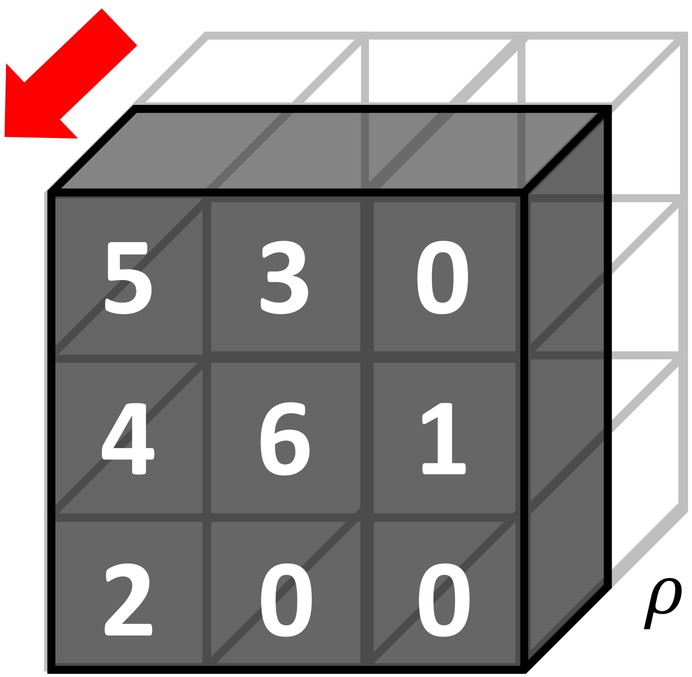


$\rho = 2.9$

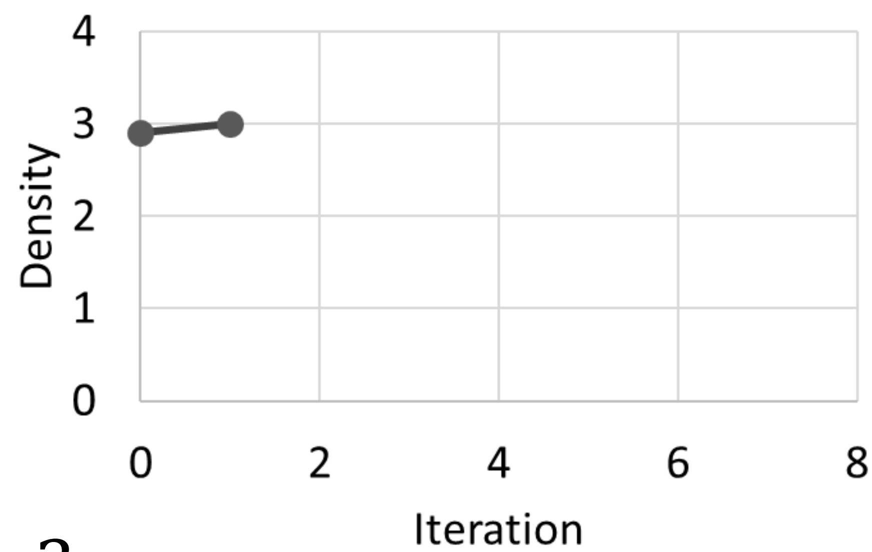


Single Dense Block Detection (cont.)

- Remove a slice to maximize density ρ

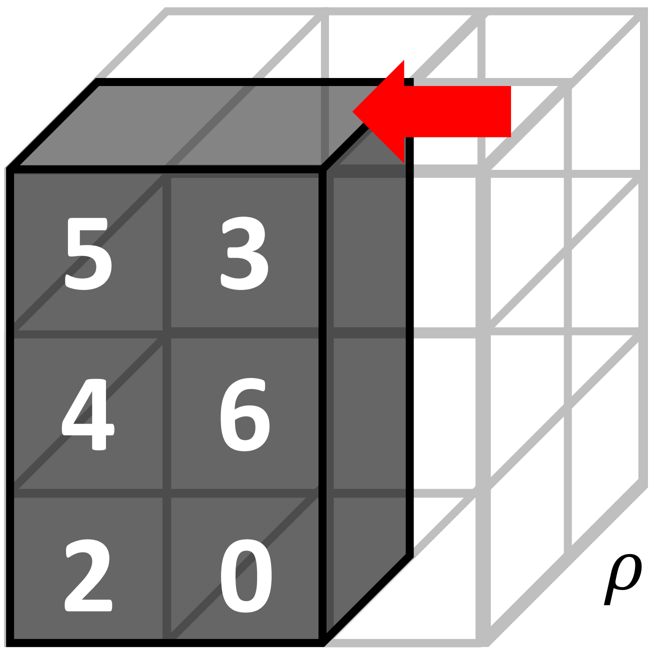


$\rho = 3$

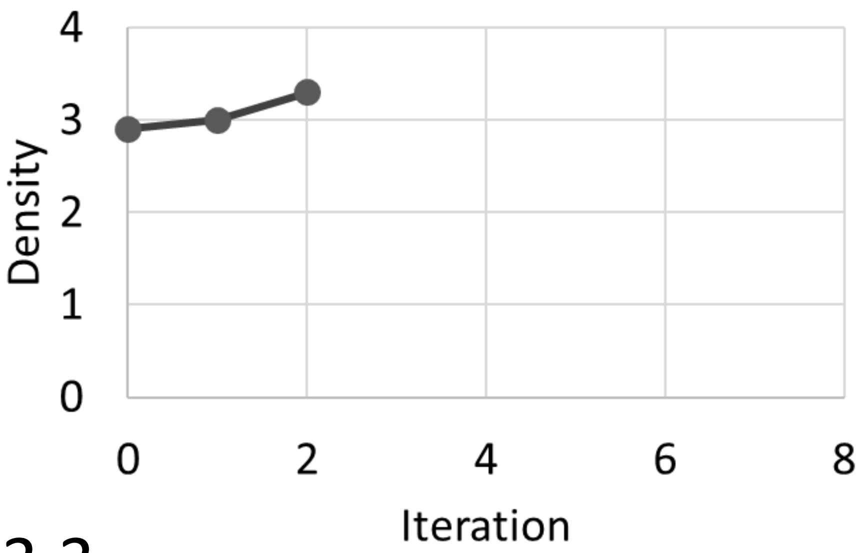


Single Dense Block Detection (cont.)

- Remove a slice to maximize density ρ

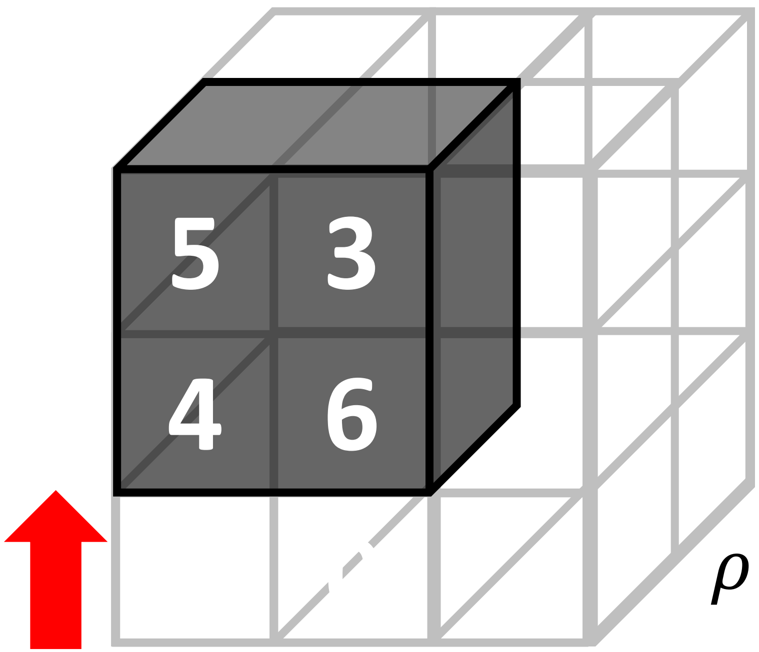


$\rho = 3.3$

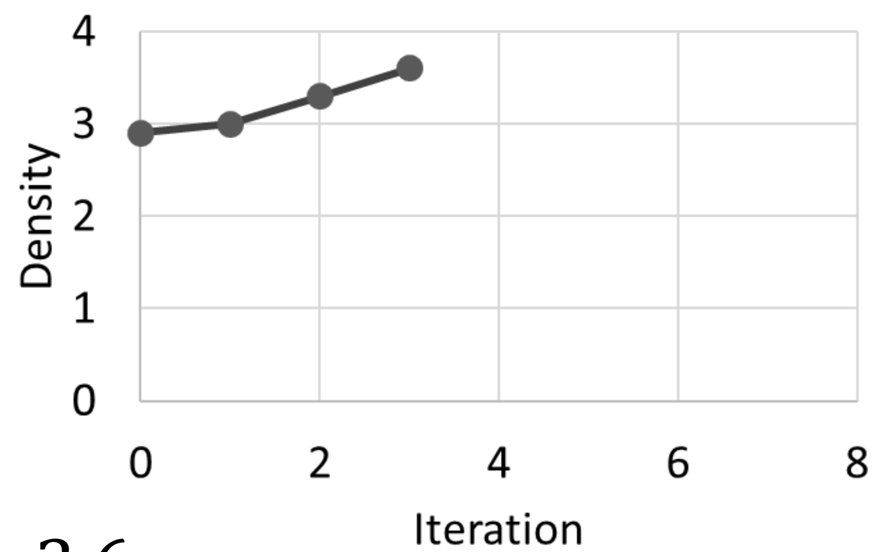


Single Dense Block Detection (cont.)

- Remove a slice to maximize density ρ

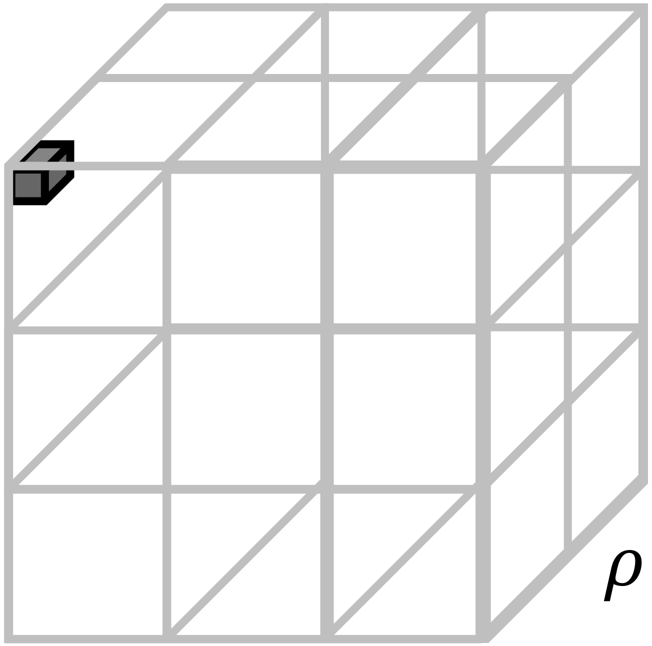


$\rho = 3.6$

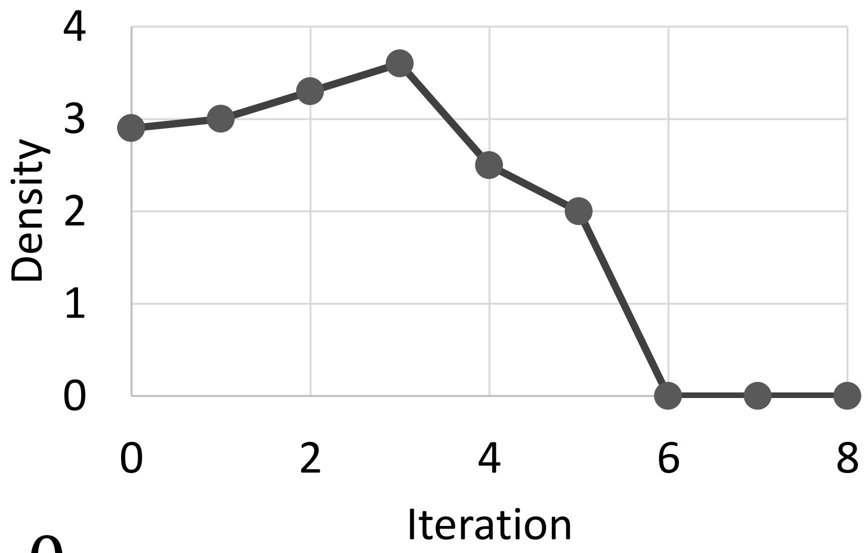


Single Dense Block Detection (cont.)

- Until all slices are removed

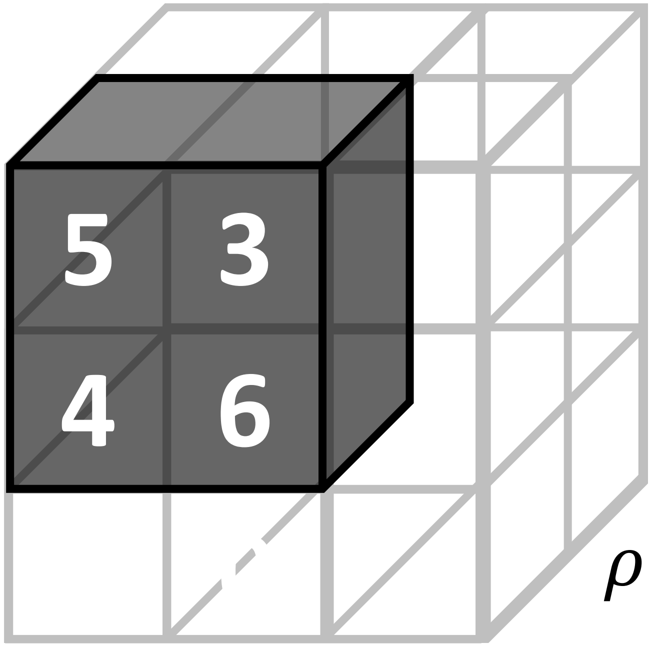


$\rho = 0$

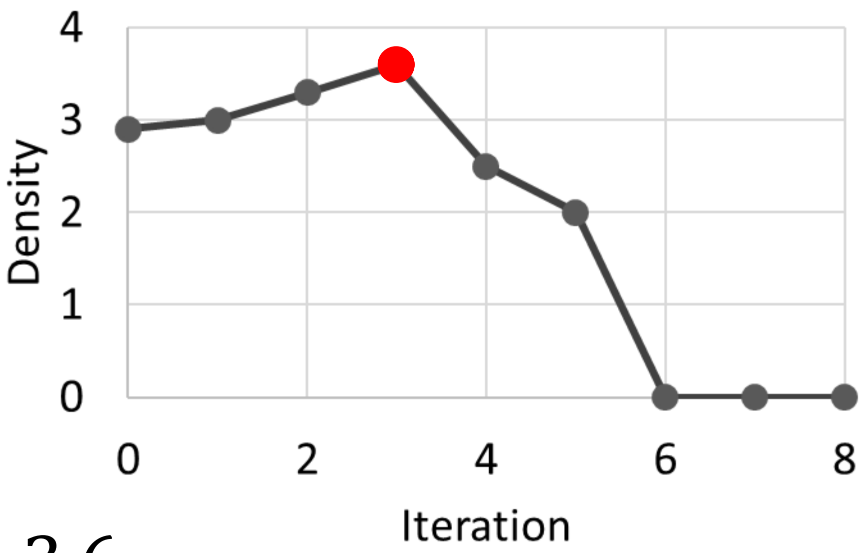


Single Dense Block Detection (cont.)

- Output: return the densest block so far



$\rho = 3.6$

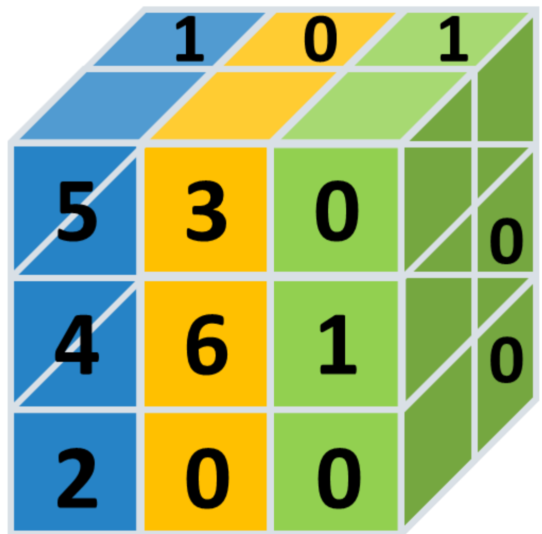


Detail

Speeding Up Process

- **Lemma 1 [Remove Minimum Sum First]**

Among slices in the same dimension, removing the slice with smallest sum of entries increases ρ most



12 > 9 > 2

Detail

Accuracy Guarantee

- **Theorem 1 [Approximation Guarantee]**

$$\rho_A(B) \geq \frac{1}{N}\rho_A(B^*)$$

M-Zoom Result

Order

Densest Block

- **Theorem 2 [Near-linear Time Complexity]**

$$O(NM\log L)$$

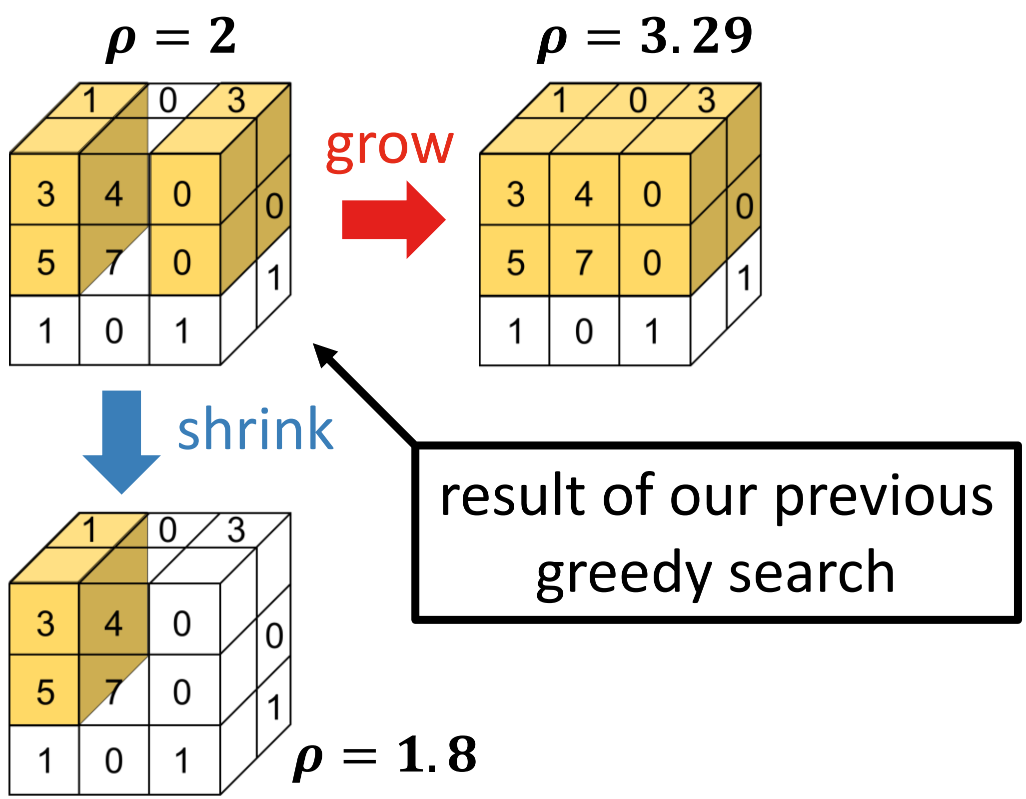
Order

Non-zeros

Entries in each mode

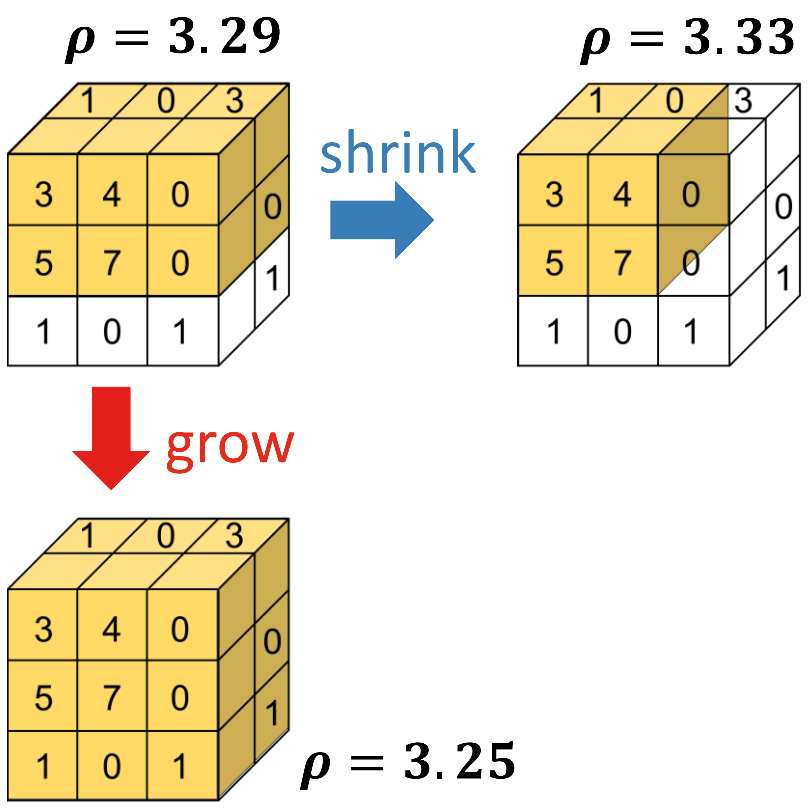
Optional Post Process

- **Local search**
  - grow or shrink until a local maximum is reached



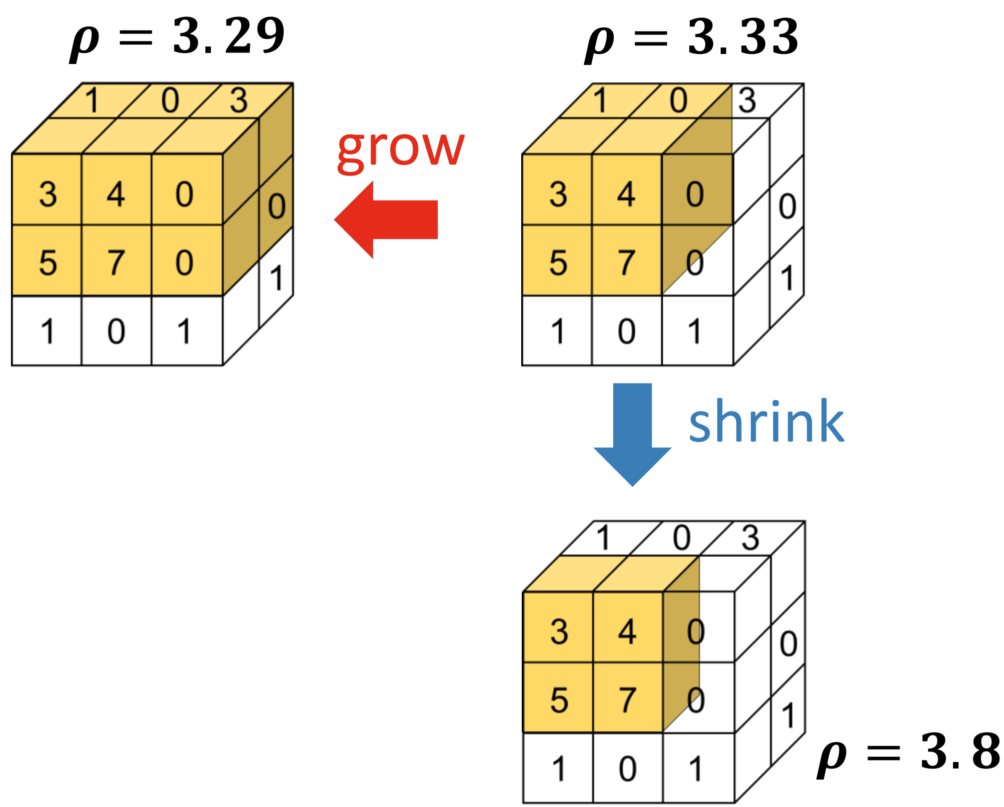
Optional Post Process (cont.)

- **Local search**
  - grow or shrink until a local maximum is reached



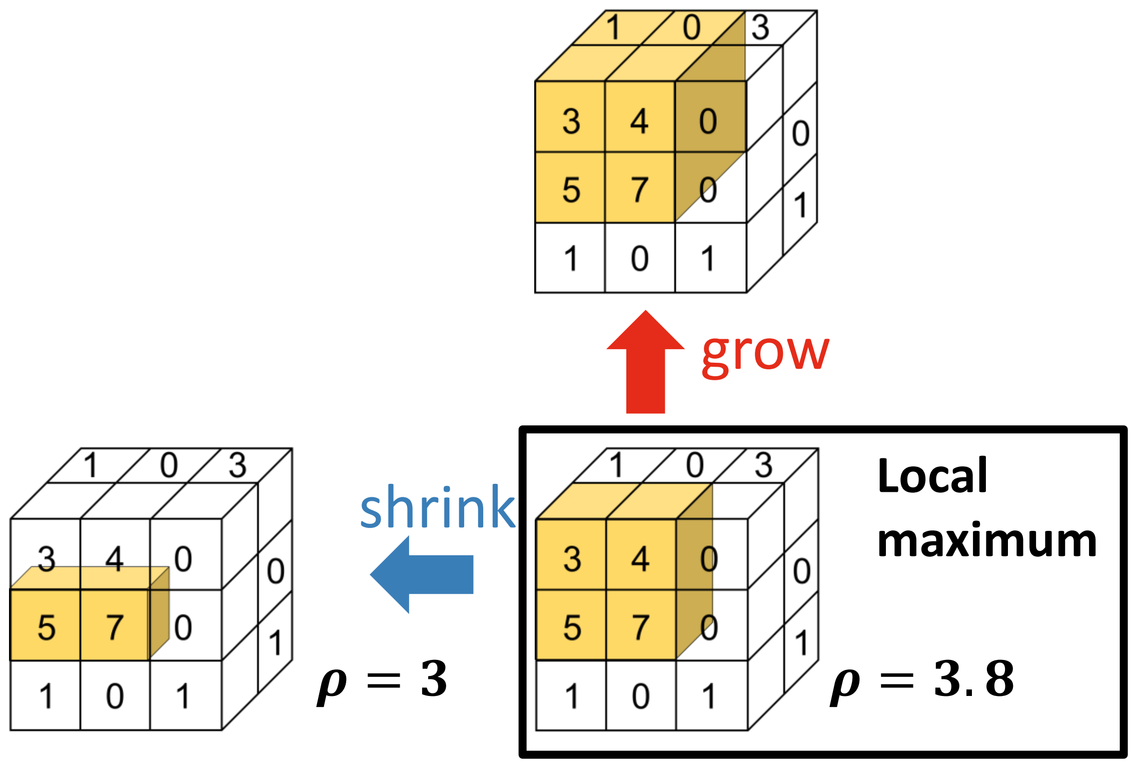
Optional Post Process (cont.)

- **Local search**
  - grow or shrink until a local maximum is reached



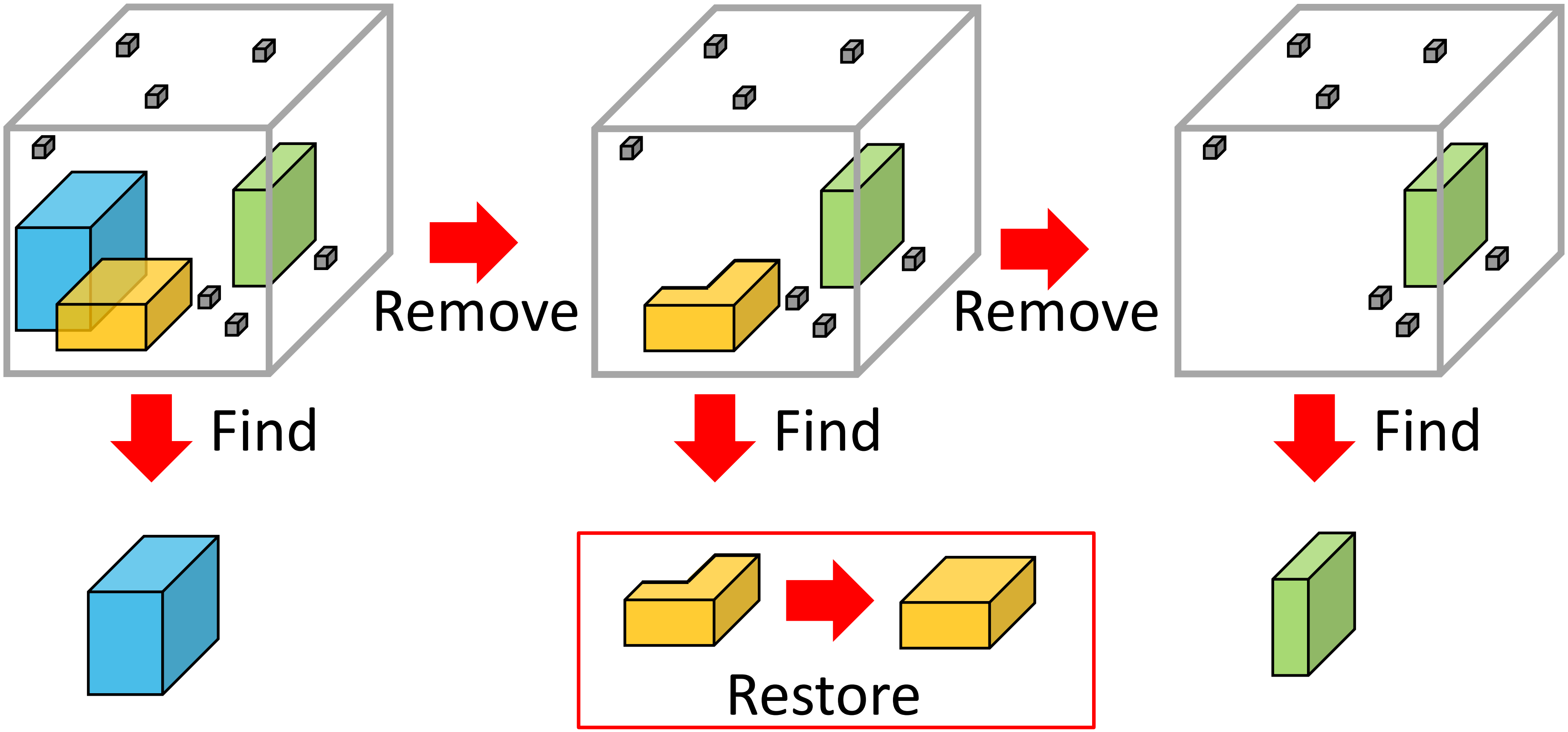
Optional Post Process (cont.)

- **Local search**
  - grow or shrink until a local maximum is reached
- Return the local maximum



Multiple Block Detection

- **Deflation**: Remove found blocks before finding others

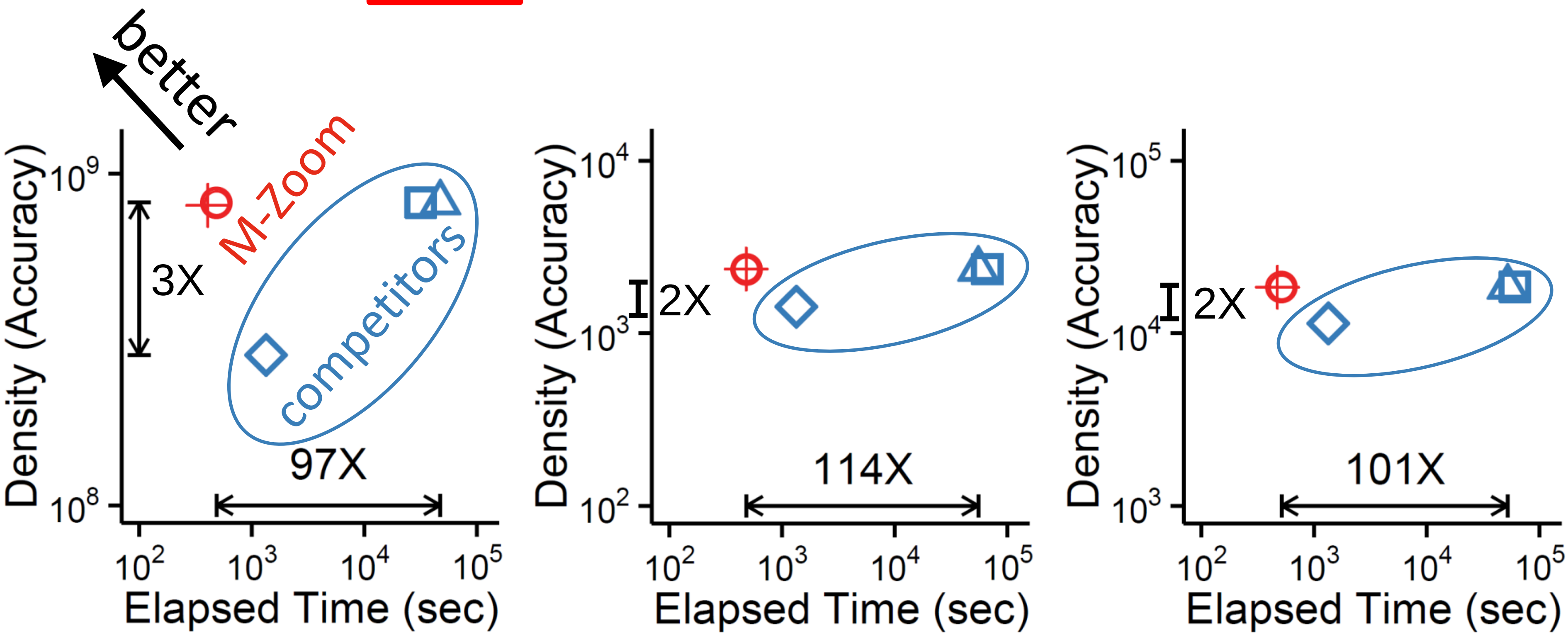


Roadmap

- Overview
- Completed Work
  - T1. Structure Analysis
  - T2. Anomaly Detection
    - T2.1 M-Zoom [PKDD 16]
      - Algorithm
      - **Experiments <<**
    - T2.2-T2.3 Related Completed Work
  - T3. Behavior Modeling
- Proposed Work
- Conclusion

Speed & Accuracy

- **Datasets:** ....

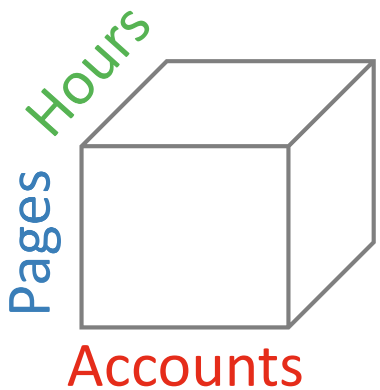


Density metric: ρ_S Density metric: ρ_A Density metric: ρ_G

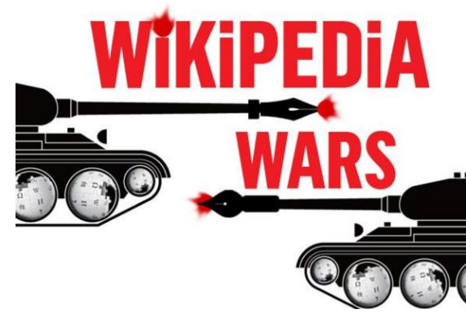
Discoveries in Practice

Korean Wikipedia

WIKIPEDIA
The Free Encyclopedia

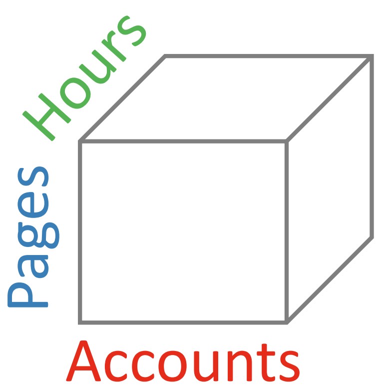


11 accounts
revised **10 pages**
2,305 times
within **16 hours**

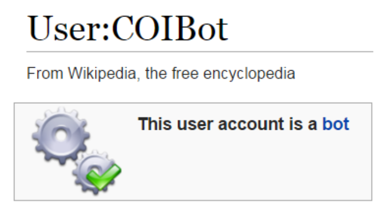


English Wikipedia

WIKIPEDIA
The Free Encyclopedia



8 accounts
revised **12 pages**
2.5 million times

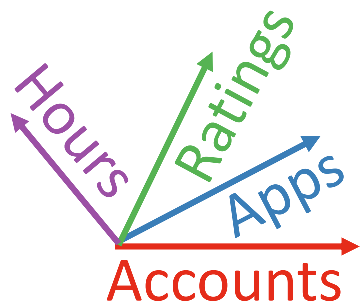


Emergency bot shutoff button

100%

Discoveries in Practice (cont.)

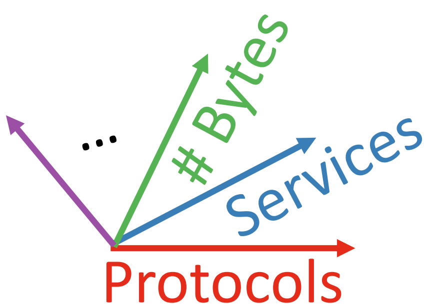
App Market
(4-order)



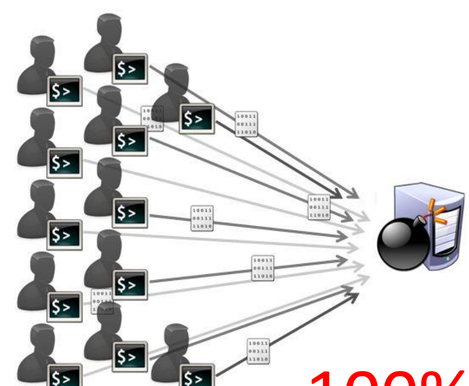
9 accounts
gives **1 product**
369 reviews with
the same rating
within **22 hours**

100%

TCP Dump
(7-order)



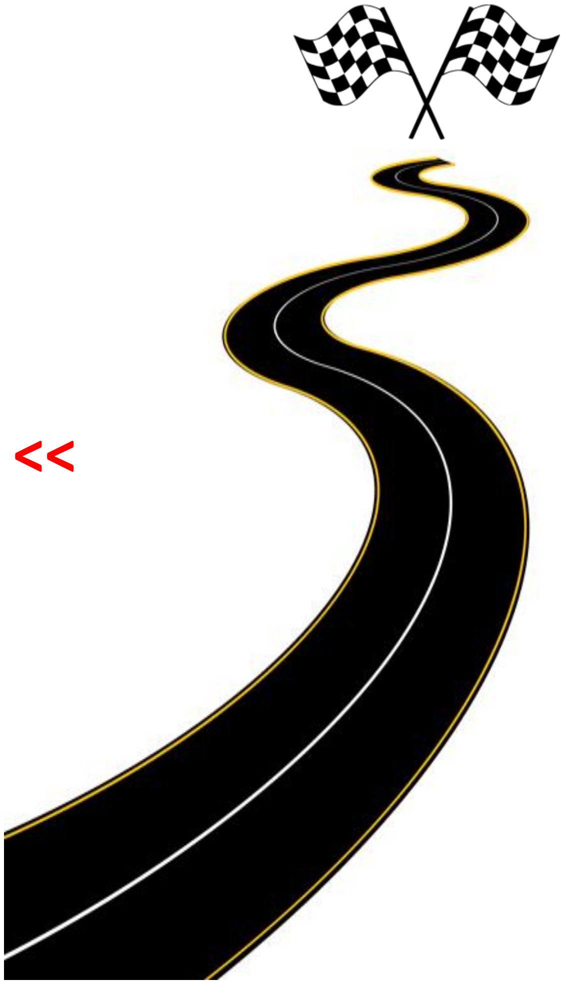
a block whose
volume = 2
and
mass = 2 millions



100%

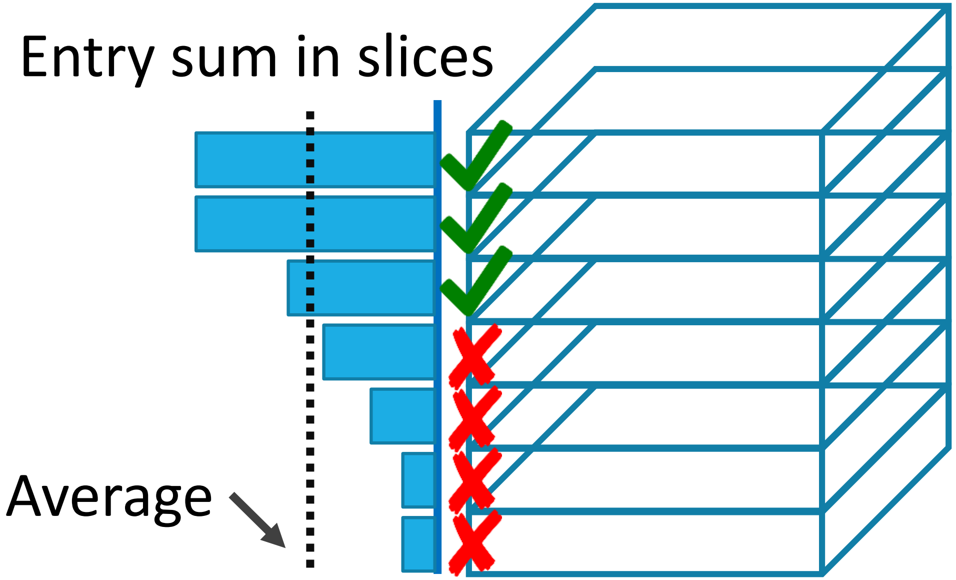
Roadmap

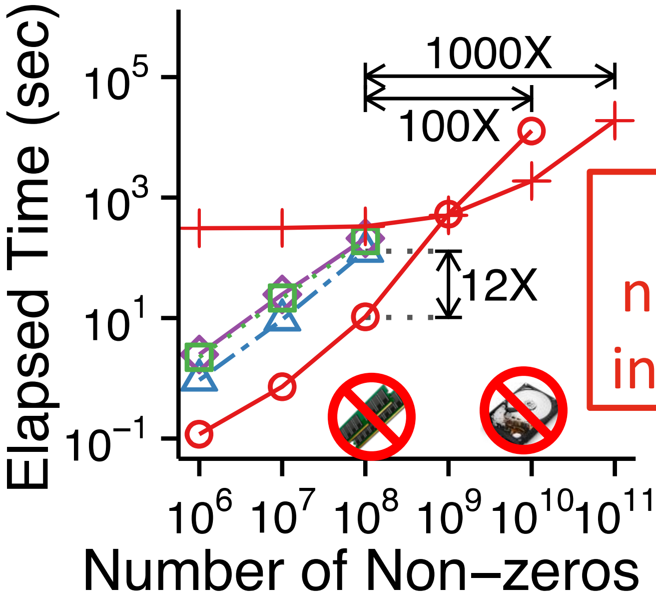
- Overview
- Completed Work
  - T1. Structure Analysis
  - T2. Anomaly Detection
    - M-Zoom
    - **T2.2-T2.3 Related Completed Work <<**
  - T3. Behavior Modeling
- Proposed Work
- Conclusion



T2.2 Extension to Web-scale Tensors

- Goal: to find dense blocks in a **disk-resident** or **distributed tensor**
- *D-Cube*: gives the **same accuracy guarantee** of M-Zoom with much **less iterations**



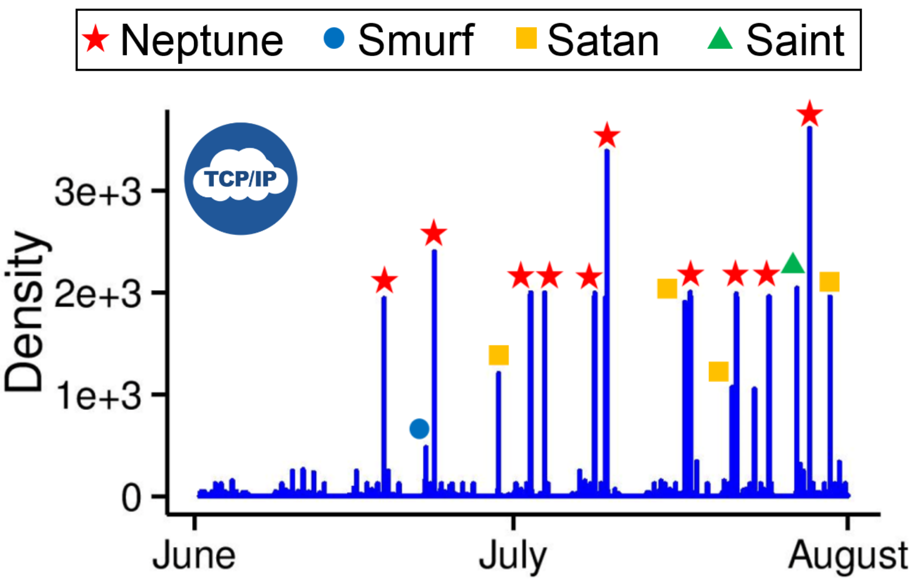


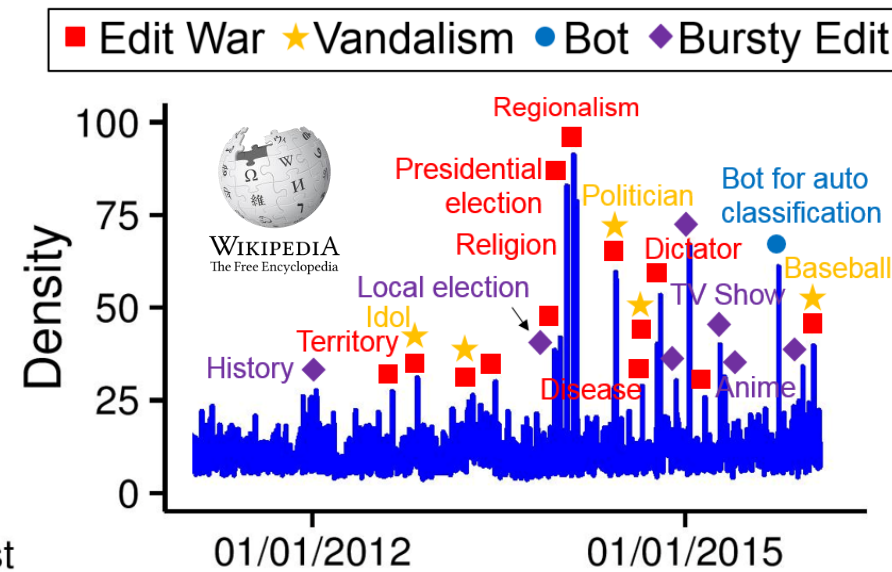
100 B nonzeros in 5 hours

Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos,
"D-Cube: Dense-Block Detection in Terabyte-Scale Tensors", WSDM 2017

T2.3 Extension to Dynamic Tensors

- Goal: to maintain a dense block in a **dynamic tensor that changes over time**
- *DenseStream*: **incrementally** computes a dense block with the **same accuracy guarantee** of M-Zoom





Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos,
"DenseAlert: Incremental Dense-Subtensor Detection in Tensor Streams", KDD 2017

Anomaly Detection in Tensors

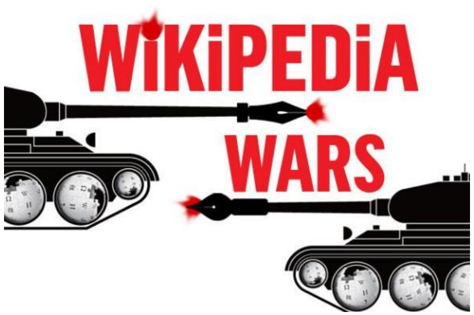
Algorithms:

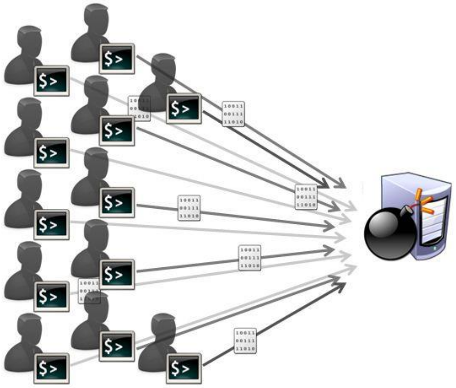
- **M-Zoom, D-Cube,** and **DenseStream**

Analyses: approximation guarantees

Discoveries:

- Edit war, vandalism, and bot activities
- Network intrusion
- Spam reviews



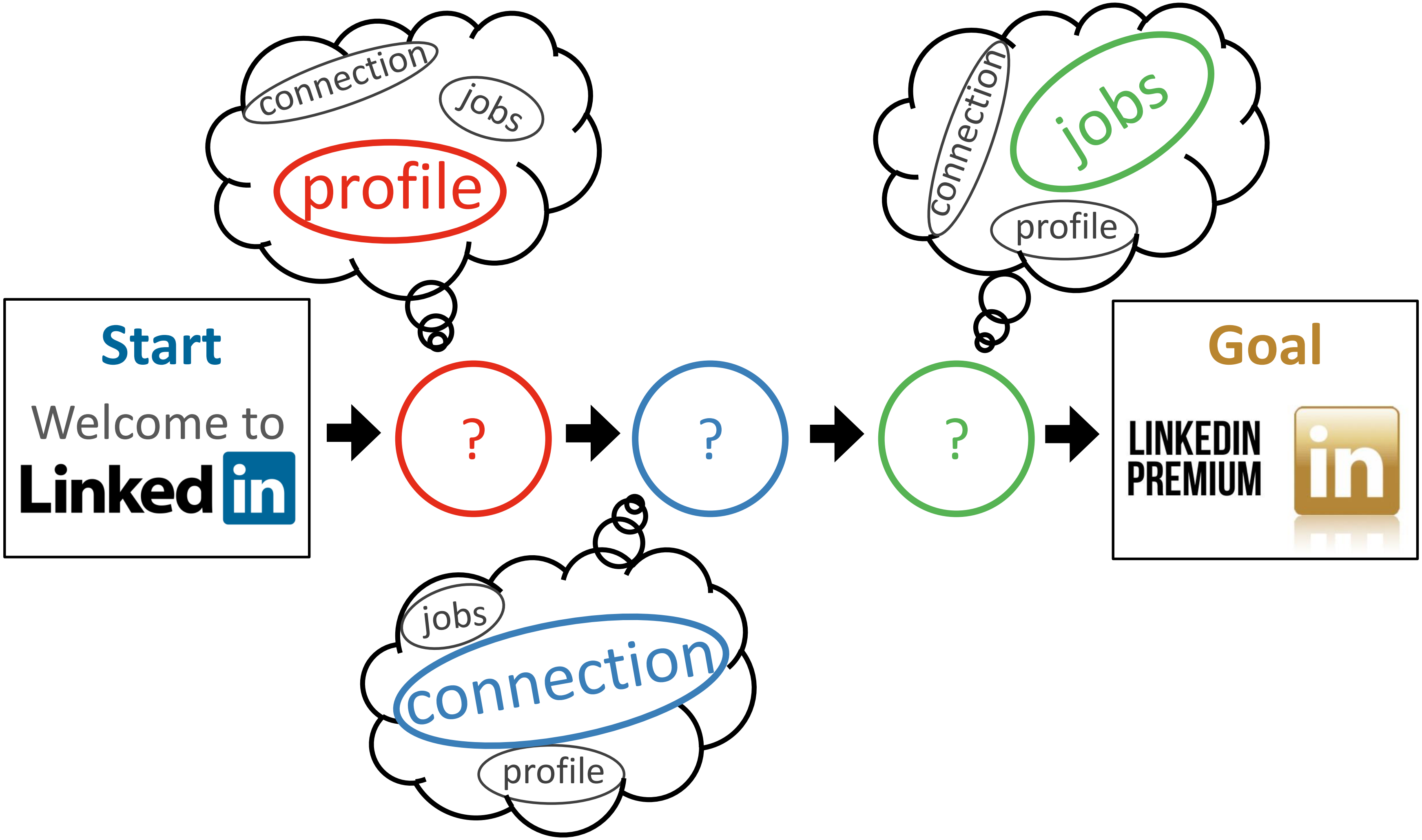


Completed Work by Topics

	T1. Structure Analysis	T2. Anomaly Detection	T3. Behavior Modeling
Graphs	Triangle Co [ICDM17][PAKD [submitted to KDD] Degenera [ICDM16]* [KAIS	skip Anomalou Subgraph [ICDM16]* [KAIS18]*	skip Purchase Behavior [IJCAI17]
Tensors	skip Summariza [WSDM17]	Dense Subten [PKDD16][WSDM1 [KDD17][TKDD18]	Progressive Behavior [WWW18]

* Duplicated

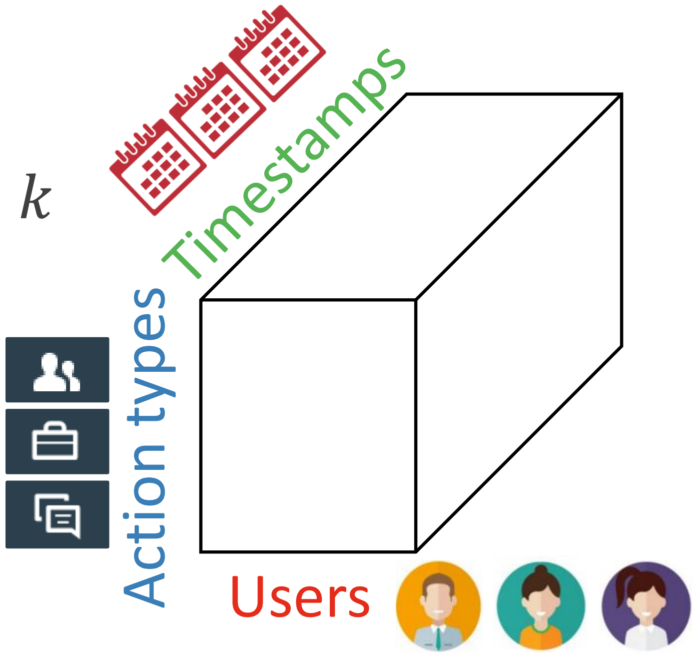
Motivation



Start
Welcome to
LinkedIn
connection
jobs
profile
?
?
?
jobs
connection
profile
connection
jobs
profile
Goal
LINKEDIN
PREMIUM

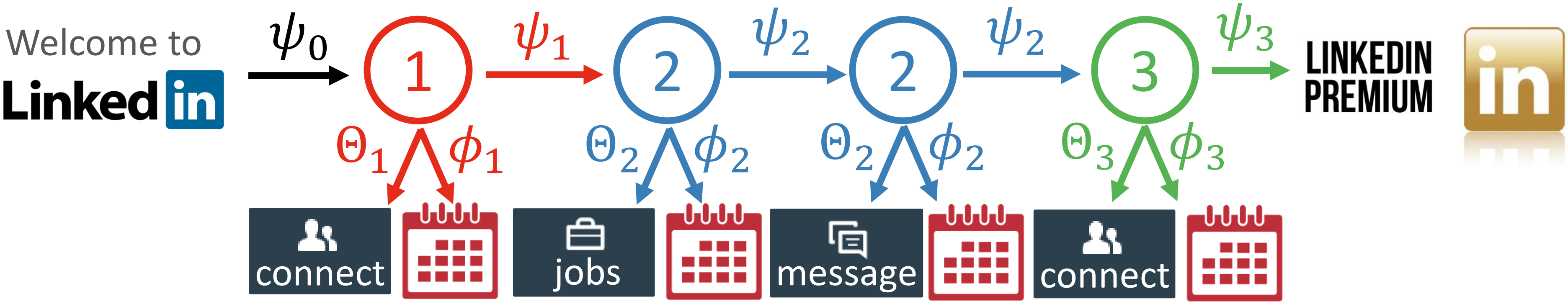
Problem Definition

- **Given**:
  - behavior log
  - number of desired latent stages: k
- **Find**: k progression stages
  - types of actions
  - frequency of actions
  - transitions to other stages
- **To best describe** the given behavior log

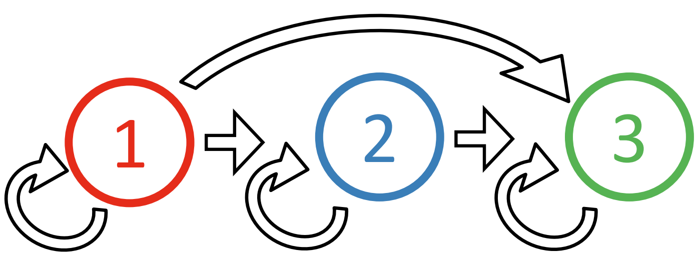


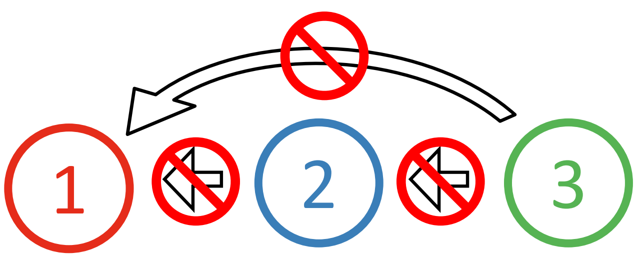
Behavior Model

- Generative process:
  - Θ_s: **action-type** distribution in stage s
  - ϕ_s: **time-gap** distribution in stage s
  - ψ_s: **next-stage** distribution in stage s



- Constraint: “no decline” (progression but no cyclic patterns)

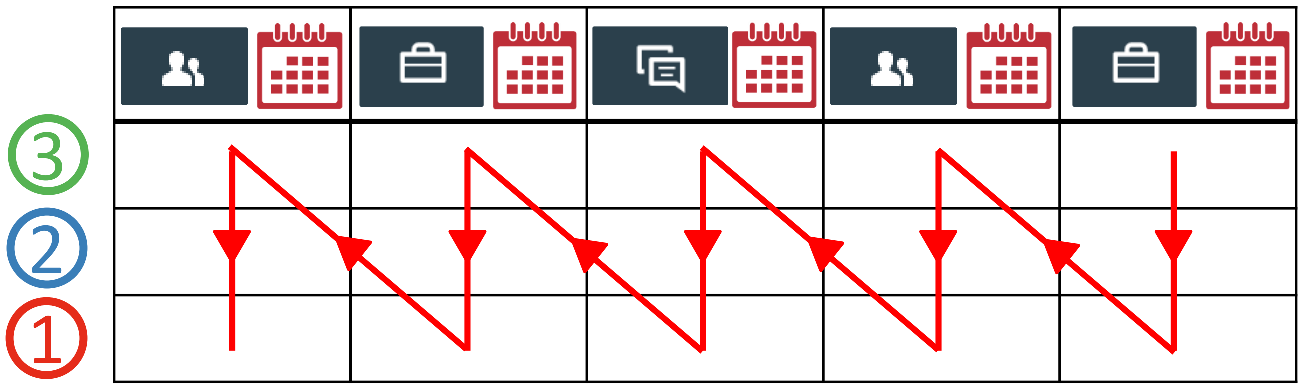




Detail

Optimization Algorithm

- **Goal:** to fit our model to given data
  - parameters: distributions (i.e., $\{\Theta_s, \phi_s, \psi_s\}_s$) and latent stages
- **repeat** until convergence
  - **assignment step**: assign latent stages while fixing prob. distributions

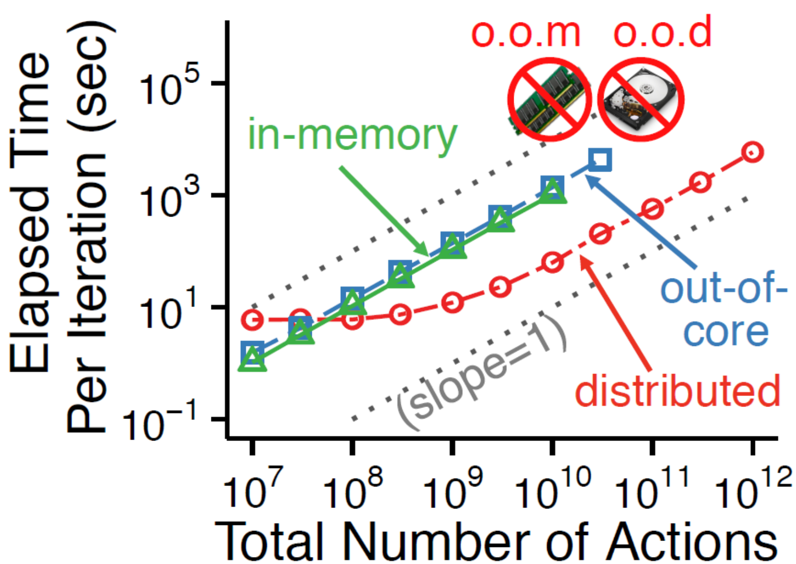


"no decline"
→ Dynamic Programming

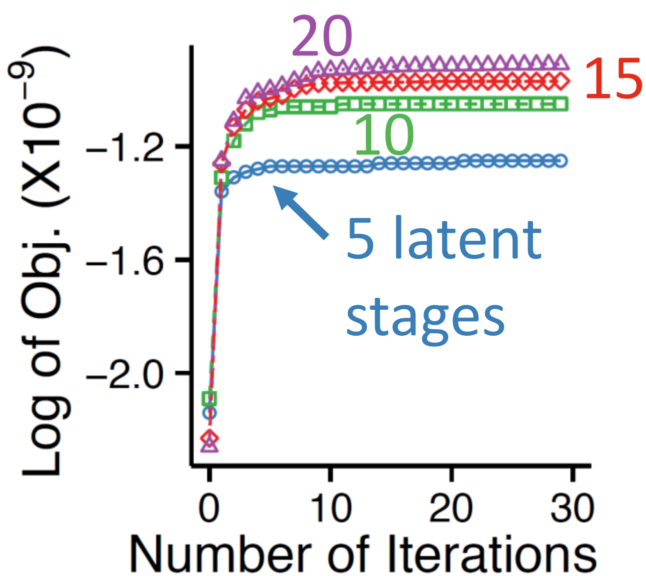
  - **update step**: update prob. distributions while fixing latent stages
    - e.g., $\Theta_s \leftarrow$ ratio of the types of actions in stage s

Scalability & Convergence

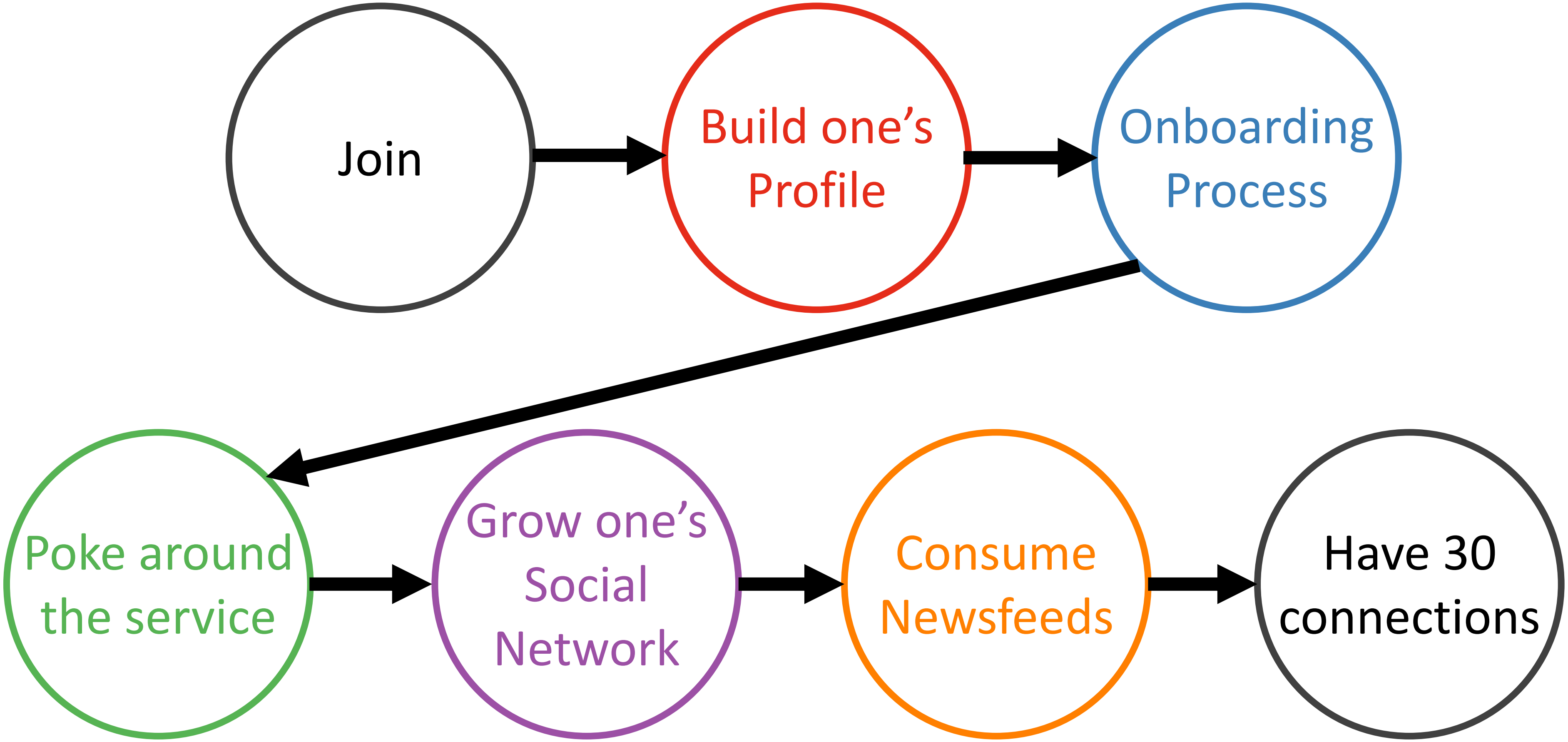
- Three versions of our algorithm
  - **In-memory**
  - **Out-of-core** (or external-memory)
  - **Distributed**



1 trillion actions in 2 hours



Progression of Users in LinkedIn



Completed Work by Topics

	T1. Structure Analysis	T2. Anomaly Detection	T3. Behavior Modeling
Graphs	Triangle Co [ICDM17][PAKD [submitted to KDD] ✔ Degenera [ICDM16]* [KAIS ✔	skip Anomalous Subgraph [ICDM16]* [KAIS18]*	skip Purchase Behavior [IJCAI17]
Tensors	Summariza [WSDM17] skip	Dense Subten [PKDD16][WSDM1 [KDD17][TKDD18] ✔	Progressi Behavior [WWW18] ✔

* Duplicated

Roadmap

- Overview
- Completed Work
    - T1. Structure Analysis
    - T2. Anomaly Detection
    - T3. Behavior Modeling
- **Proposed Work <<**
- Conclusion

Proposed Work by Topics

	T1. Structure Analysis	T2. Anomaly Detection	T3. Behavior Modeling
Graphs	P1. Triangle Counting in Fully Dynamic Stream		P3. Polarization Modeling
Tensors	P2. Fast and Scalable Tucker Decomposition		

* Duplicated

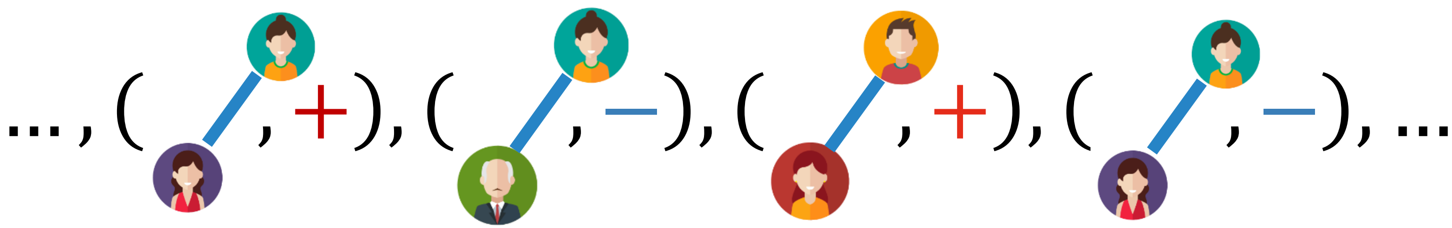
Proposed Work by Topics

	T1. Structure Analysis	T2. Anomaly Detection	T3. Behavior Modeling
Graphs	P1. Triangle Counting in Fully Dynamic Stream		P3. Polarization Modeling
Tensors	P2. Fast and Scalable Tucker Decomposition		

* Duplicated

P1: Problem Definition

- Given:
  - a **fully dynamic** graph stream,
    - i.e., list of edge insertions and edge deletions



  - Memory budget k
- Estimate: the **counts of global and local triangles**
- To Minimize: estimation error

P1: Goal

Method	Accuracy	Handle Deletions?
Triest-FD	Lowest	Yes
MASCOT	Low	No
Triest-IMPR	High	No
WRS	Highest	No
Proposed	**Highest**	**Yes**

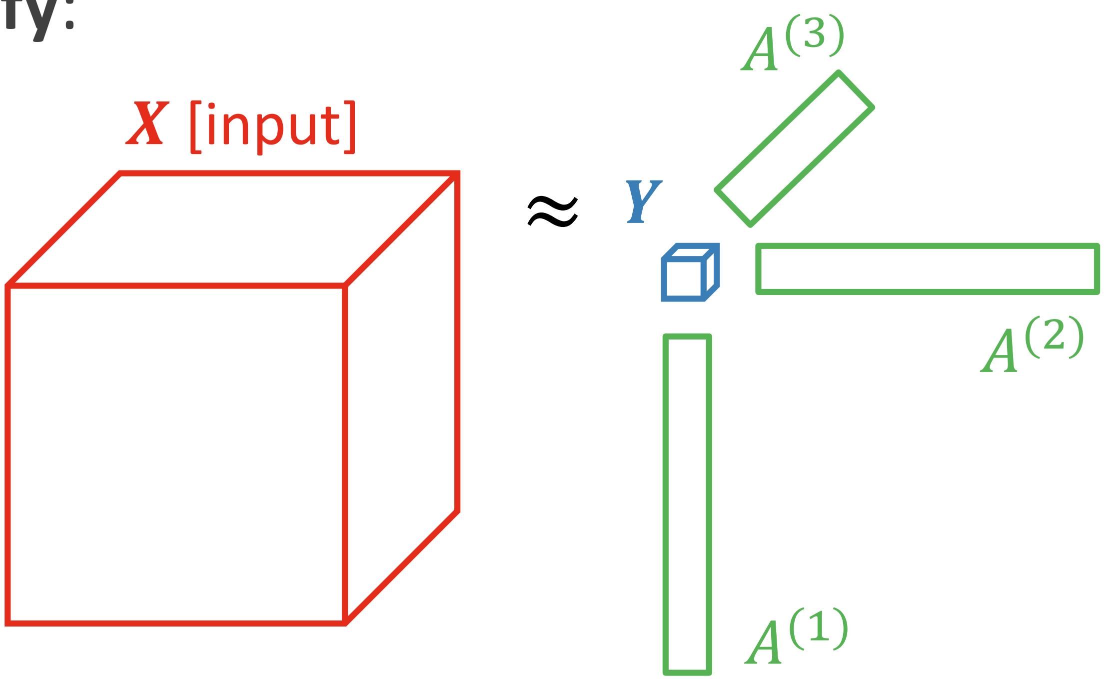
Proposed Work by Topics

	T1. Structure Analysis	T2. Anomaly Detection	T3. Behavior Modeling
Graphs	P1. Triangle Counting in Fully Dynamic Stream		P3. Polarization Modeling
Tensors	P2. Fast and Scalable Tucker Decomposition		

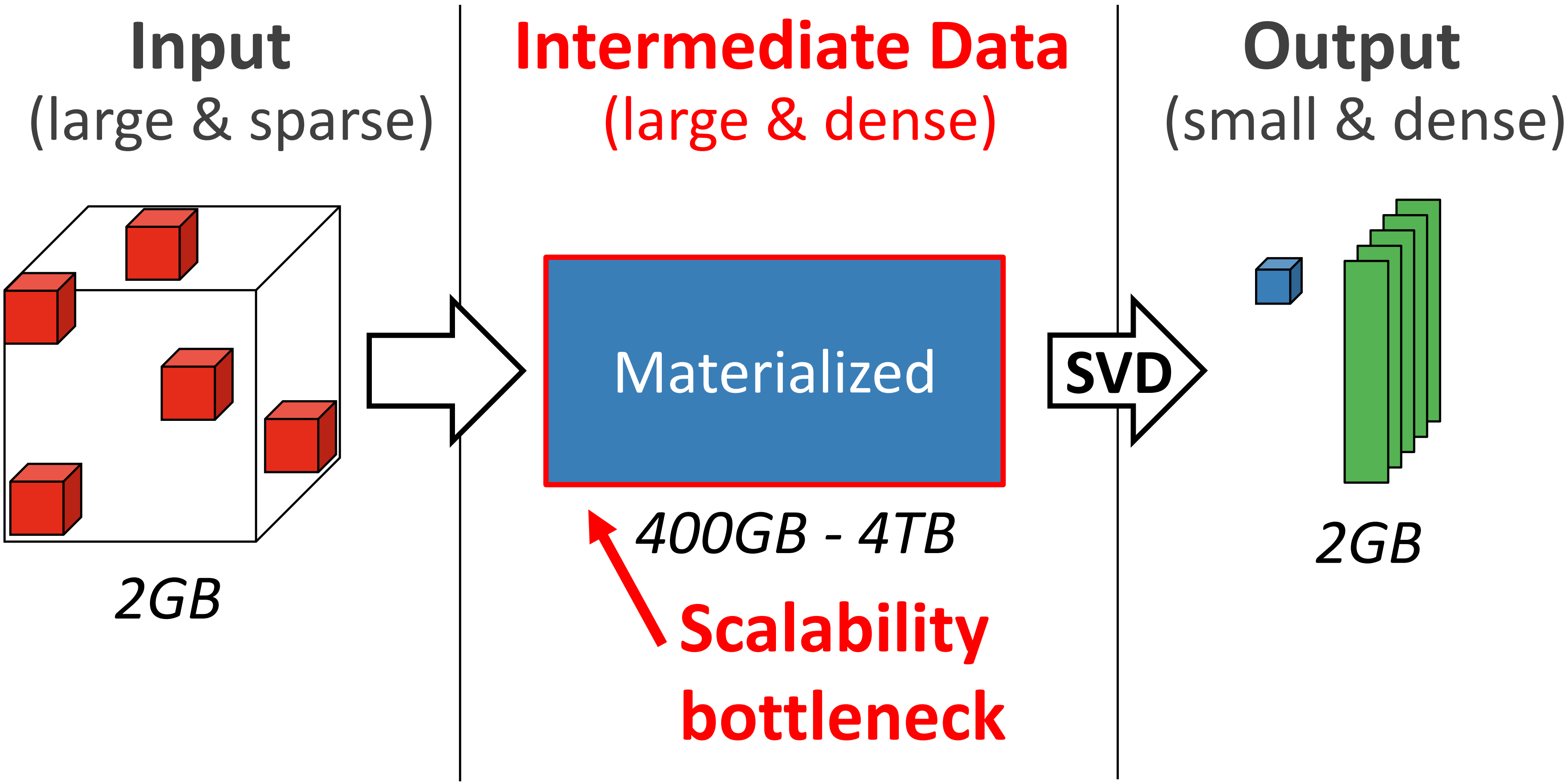
* Duplicated

P2: Problem Definition

- Tucker Decomposition (a.k.a High-order PCA)
  - **Given**: an N-order input tensor $\boldsymbol{X}$
  - **Find**: N factor matrices $A^{(1)} \ldots A^{(N)}$ & core-tensor $\boldsymbol{Y}$
  - **To satisfy**:

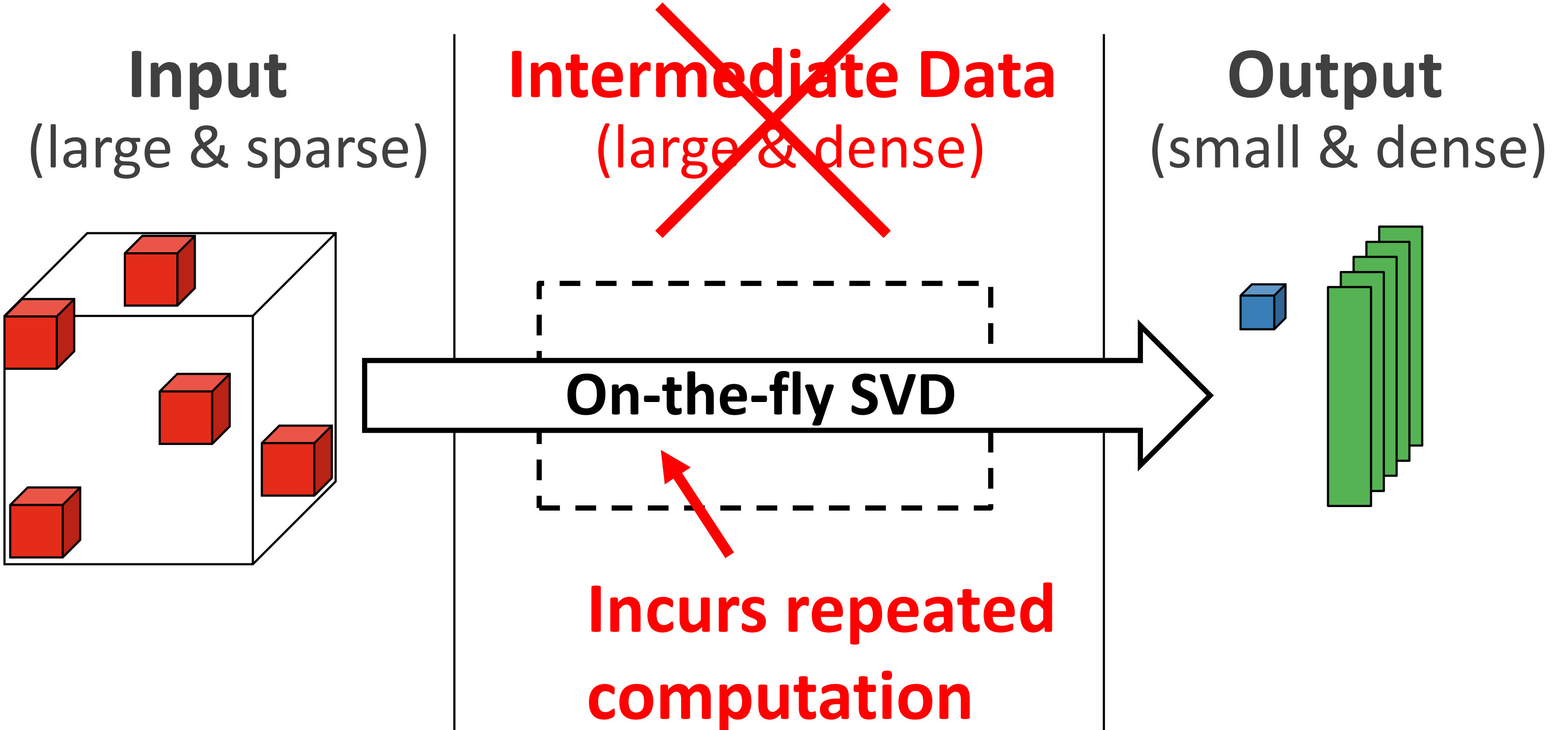


P2: Standard Algorithms



P2: Completed Work

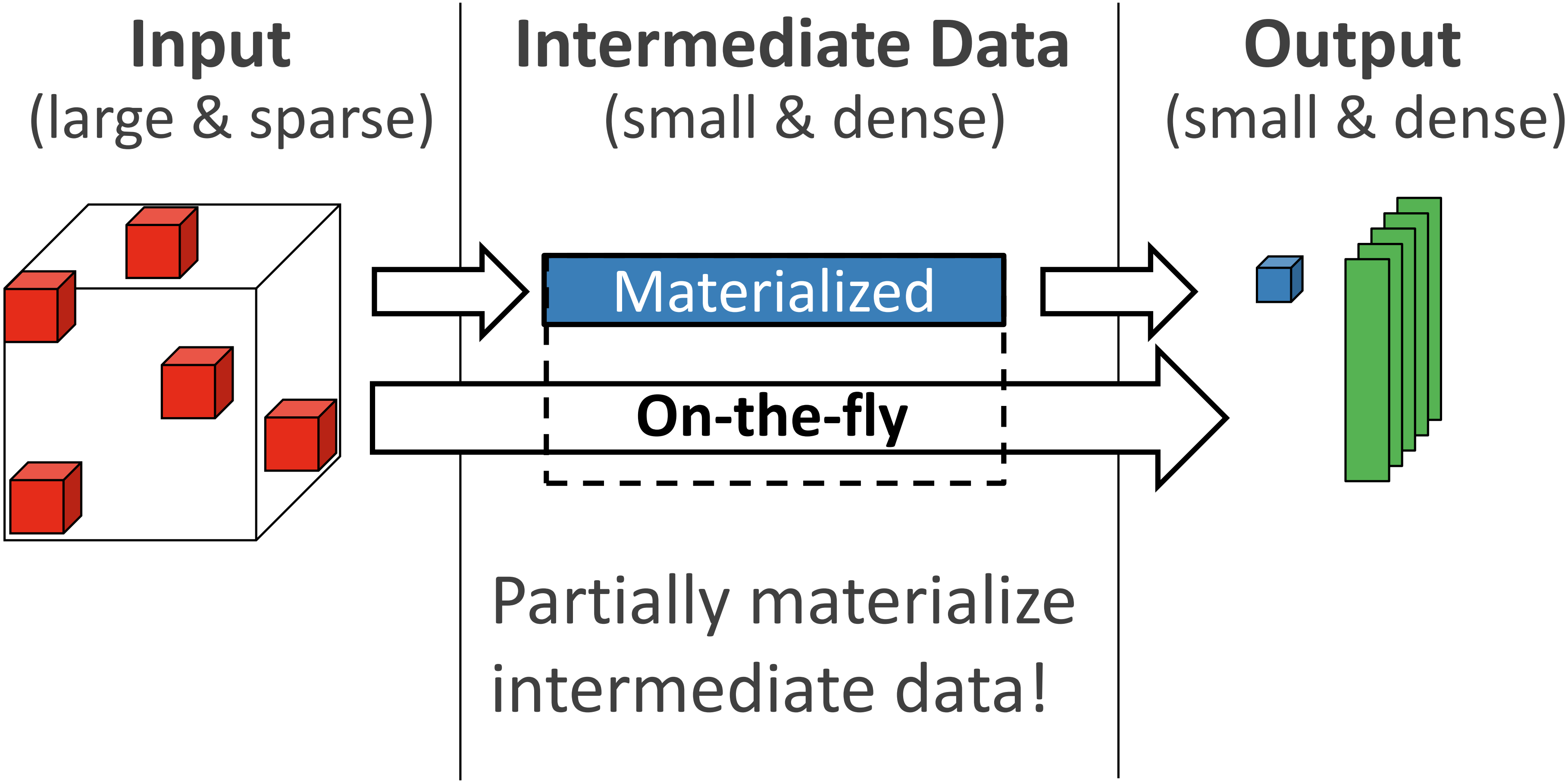
- Our completed work [WSDM17]



Jinoh Oh, **Kijung Shin**, Evangelos E. Papalexakis, Christos Faloutsos, and Hwanjo Yu, "S-HOT: Scalable High-Order Tucker Decomposition", WSDM 2017.

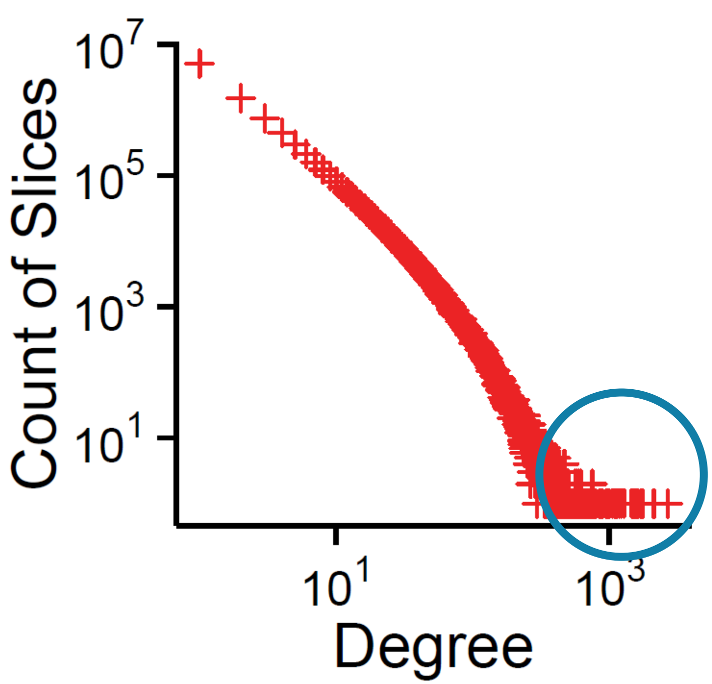
P2: Proposed Work

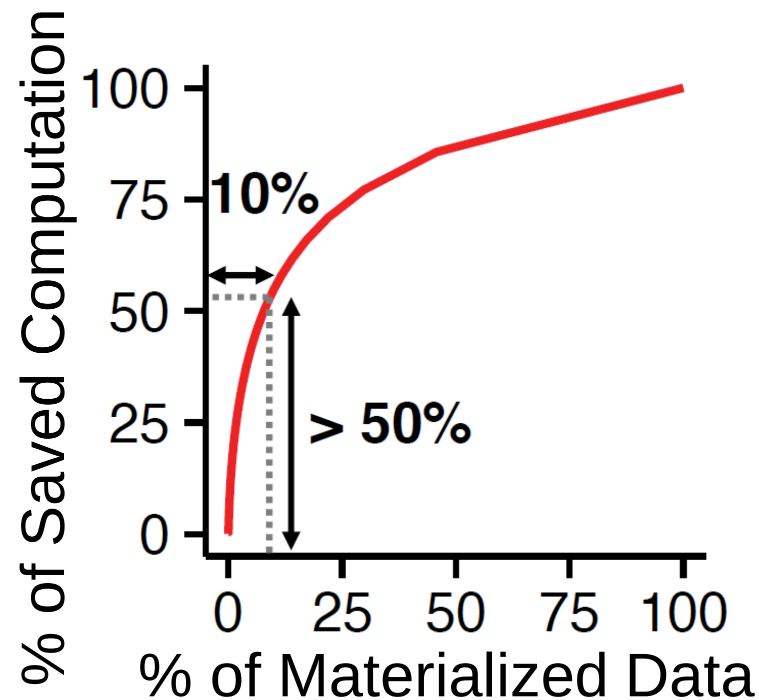
- Proposed algorithm



P2: Expected Performance Gain

- Which part of intermediate data should we materialize?
- Exploit skewed degree distributions!





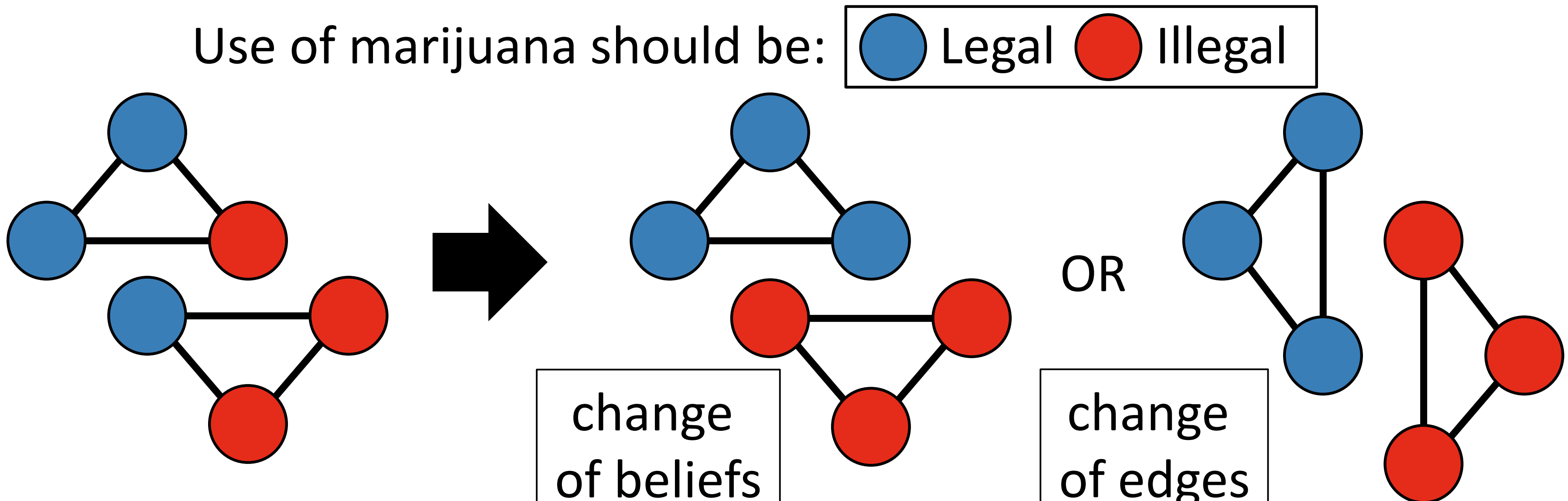
Proposed Work by Topics

	T1. Structure Analysis	T2. Anomaly Detection	T3. Behavior Modeling
Graphs	P1. Triangle Counting in Fully Dynamic Stream		P3. Polarization Modeling
Tensors	P2. Fast and Scalable Tucker Decomposition		

* Duplicated

P3. Polarization Modeling

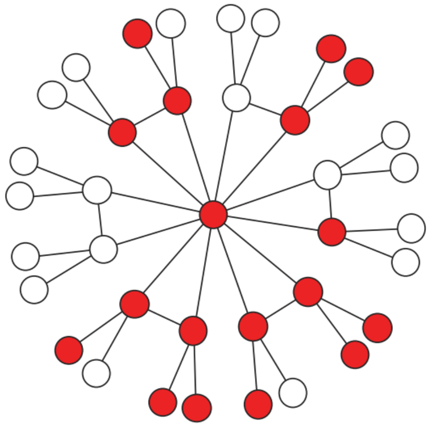
- **Polarization** in social networks: division into contrasting groups



"How do people choose between two ways of polarization?"

P3. Problem Definition

- **Given**: time-evolving social network with nodes' beliefs on controversial issues
  - e.g., legalizing marijuana
- **Find**: actor-based model with a utility function
  - depending on network features, beliefs, etc.
- **To best describe:** the polarization in data
- **Applications**:
  - predict future edges
  - predict the cascades of beliefs



Proposed Work by Topics

	T1. Structure Analysis	T2. Anomaly Detection	T3. Behavior Modeling
Graphs	P1. Triangle Counting in Fully Dynamic Stream		P3. Polarization Modeling
Tensors	P2. Fast and Scalable Tucker Decomposition		

* Duplicated

Timeline

- Mar-May 2018
  - **P1**. Triangle counting in fully dynamic graph streams
- Jun-Aug 2018
  - **P3**. Polarization modeling
- Sep-Oct 2018
  - **P2**. Fast and scalable tucker decomposition
- Nov 2018 –April 2019
  - Thesis Writing & Job Application
- May 2019
  - Defense

Roadmap

- Overview
- Completed Work
  - T1. Structure Analysis
  - T2. Anomaly Detection
  - T3. Behavior Modeling
- Proposed Work
- **Conclusion <<**

Conclusion

- **Goal**:

  To Understand Large Dynamic Graphs and Tensors

- **Subtasks**:
  - structure analysis 
  - anomaly detection 
  - behavior modeling 
- **Approaches**:
  - distributed or external-memory algorithms
  - streaming algorithms based on sampling
  - approximation algorithms

References (Completed work)

[1] **Kijung Shin**, Bryan Hooi, and Christos Faloutsos, "M-Zoom: Fast Dense-Block Detection in Tensors with Quality Guarantees", ECML/PKDD 2016

[2] **Kijung Shin**, Tina Eliassi-Rad, and Christos Faloutsos, "CoreScope: Graph Mining Using k-Core Analysis - Patterns, Anomalies and Algorithms", ICDM 2016

[3] **Kijung Shin**, "Mining Large Dynamic Graphs and Tensors for Accurate Triangle Counting in Real Graph Streams", ICDM 2017

[4] Jinoh Oh, **Kijung Shin**, Evangelos E. Papalexakis, Christos Faloutsos, and Hwanjo Yu, "S-HOT: Scalable High-Order Tucker Decomposition", WSDM 2017

[5] **Kijung Shin**, Bryan Hooi, Jisu Kim, and Christos Faloutsos, "D-Cube: Dense-Block Detection in Terabyte-Scale Tensors", WSDM 2017

[6] **Kijung Shin**, Euiwoong Lee, Dhivya Eswaran, and Ariel D. Procaccia, "Why You Should Charge Your Friends for Borrowing Your Stuff", IJCAI 2017

[7] **Kijung Shin**, Bryan Hooi, Jisu Kim, and Christos Faloutsos, "DenseAlert: Incremental Dense-Subtensor Detection in Tensor Streams", KDD 2017

[8] **Kijung Shin**, Bryan Hooi, and Christos Faloutsos, "Fast, Accurate and Flexible Algorithms for Dense Subtensor Mining", TKDD 2018

[9] **Kijung Shin**, Tina Eliassi-Rad, and Christos Faloutsos, "Patterns and Anomalies in k-Cores of Real-world Graphs with Applications", KAIS 2018

[10] **Kijung Shin**, Mahdi Shafiei, Myunghwan Kim, Aastha Jain, and Hema Raghavan, "Discovering Progression Stages in Trillion-Scale Behavior Logs", WWW 2018

[11] **Kijung Shin**, Mohammad Hammoud, Euiwoong Lee, Jinoh Oh, and Christos Faloutsos. "Kijung Shin, Mohammad Hammoud, Euiwoong Lee, Jinoh Oh, and Christos Faloutsos. PAKDD 2018." PAKDD 2018

Thank You

- Papers, software, data: http://www.cs.cmu.edu/~kijungs/proposal/
- Email: kijungs@cs.cmu.edu
- Thanks to:
  - Sponsors:
  - Admins:

  - Collaborators: