Summarizing Graphs at Multiple Scales: New Trends

Danai Koutra
University of Michigan

Jilles Vreeken
CISPA Helmholtz Center for Information Security

Francesco Bonchi
ISI Foundation
Roadmap

- 1:30-1:45pm  Introduction  [Jilles]
- 1:45-2:50pm  Network-level Summaries  [Francesco]
- 2:55-3:20pm  Multi-network Summaries  [Danai]
- 3:20-3:40pm  break
- 3:40-4:05pm  Multi-network Summaries  [Danai]
- 4:10-4:40pm  Node-level Summaries  [Jilles]
- 4:40-4:50pm  Conclusion  [Jilles]
Part II: Multinetwork-level Summaries
Static vs. Time-evolving graph

- Adjacency matrix $A$

  Static graph

- 3D matrix (tensor)

  Dynamic, temporal, or time evolving graph
Dynamic Graph Summarization: Definition

- **Input**: dynamic graph $G$
- **Output:**
  - a temporal summary graph or
  - a set of possibly overlapping structures
- to *concisely describe* the given graph
Not explored much

Challenges

• methods sensitive to time granularity (often chosen arbitrarily)
• continuous / irregular change of real-world graphs
• online “interestingness” measure
• visualization
Approach 1:
- treat a dynamic graph as a series of static graphs
- apply static graph summarization methods

Shortcomings:
- what is the right time granularity for the snapshots?
  - too short: a lot of data processing
  - too long: miss patterns (e.g., bursty behavior)
- how to “link” the static summaries?
Basic approaches for handling dynamic graphs

Approach 2:
- create an aggregate / approximation graph
  - recency / frequency of interactions
  - aggregated edge weights via kernel smoothing
    - exponential, inverse linear, linear, uniform
- apply static graph summarization methods

Shortcomings:
- what is the right time granularity for the snapshots?
- how to choose a kernel?
- does not capture the dynamics of the graph
Dynamic Graph Summarization

Graph Summarization

- Network type
- Static
  - Plain
  - Labeled
- Dynamic
  - Plain

Core techniques employed
- Structure only
  - Grouping
  - Compression
  - Simplification
  - Influence
- Structure + labels
  - Grouping
  - Compression
  - Influence
- Temporal structure
  - Grouping
  - Compression
  - Influence

Applications
- Query Efficiency
- Compression
- Visualization
- Pattern Discovery
- Influence Analysis

Open Problems
- Richer Data
- Streaming / Incremental Summarization
- Automated Insight Extraction
- Evaluation

Related Research Areas:
- Graph clustering, partitioning, community detection, sampling, sparsification, sketches, compression
Grouping-based Summarization

These methods group nodes into supernodes and connect them with superedges, resulting in a supergraph.
Compression-based Summarization

Given: a series of $n$ graph snapshots $G_1, G_2, \ldots, G_n$

Goal: Find a concise summary of recurrent, possibly overlapping subgraphs $G_1 \cup G_2 \cup \ldots \cup G_n$

- **Constant near-clique** in Yahoo IM
- **Periodic star** in a phonecall network
- **Ranged near-clique** in co-authorship

**Goal:** Find a concise summary of recurrent, possibly overlapping subgraphs

- scalable
- parameter free

[Shah et al., '15]
## Compression-based Summarization

1) Use a dictionary of **temporal vocabulary**:
   * Static vocabulary
   * Temporal vocabulary

2) Get the shortest lossless description (MDL)
   - better compression \(\rightarrow\) better summary

[Shah et al., '15]
Compression-based Summarization

Given: a dynamic graph $G$

temporal templates $\Phi$,

Find: the smallest model $M$

$s.t. \min L(G,M) = L(M) + L(E)$
Step 1: Generate static subgraph instances

- using VoG [Koutra et al. ‘14]

[Shah et al., ‘15]
Compression-based: TIMECRUNCH

Step 1: Generate static subgraph instances
Step 2: Stitch *static instances to temporal instances*

- **idea**: choose the patterns that compress best
- using MDL + clustering (rank-1 SVD, cosine similarity)

![Diagram showing G1, G2, G3 with bc, st, fc patterns and their compression results]

[Shah et al., '15]
Compression-based: TIMECRUNCH

Step 1: Generate static subgraph instances
Step 2: Stitch *static instances* to *temporal instances*
Step 3: Compose the dynamic graph summary
  - best summary: combinatorial
  - greedy heuristic: include temporal instances in decreasing order of benefit

Summary

- `constant bc`
- `flickering st`
- `constant fc`

[Shah et al., '15]
Patterns in Honey-net

Attacker-victim bipartite network (372K nodes)

- 71% of attacks on 12/31 – 1/1
  - “new year” exploits: “oneshot stars”

“Ranged star” attack on 589 honeypot machines lasting 2 weeks

[Shah et al., ‘15]
Patterns in Instant Messaging

- 100K users
- 2.1M message exchanges
- April 2008

“Constant near-clique” of 40 users with 55% density
- large group chat, or botnet?

[Shah et al., ‘15]
Patterns in Phonecall Graph

Who-calls-whom activity of 6.3M inhabitants of large Asian city in Dec. 2007

Oneshot near-bipartite core of 792 callers on Dec. 31

“handshake” calls between well-wishers and receivers?

[Shah et al., ‘15]
Extends GraSS to dynamic graphs

- **dynamic graph** = tensor with one dimension increasing in time
- potentially **infinite stream** of static graphs
- define a sliding tensor window
  summarize the tensor within the tensor window

[Tsalouchidou, ‘16]
Overview and contributions

At each time-stamp:
1. new adjacency matrix arrives
2. sliding window is updated (one adjacency matrix exits the window)
3. summary is created for the current window, by clustering nodes to create supernodes (following Riondato et al.)
4. output: one summary at every time-stamp

Contributions:
- two online algorithms for summarizing dynamic, large-scale graphs
- distributed, scalable algorithms, implemented in Apache Spark

[Tsalouchidou, '16]
Algorithms

Baseline:
• standard $k$-means clustering at each timestamp
• $N$ points each with $wN$ values
• observation: $(w - 1)N^2$ unchanged at every new timestamp

Two-level clustering:
• adjacency matrix to micro-clusters
• keep statistics in the micro-clusters
• run maintenance algorithm
• micro-clusters to supernodes

[Tsalouchidou, '16]
**Idea:** TCM

- creates graph **sketches**
- approximates graph queries by querying \( d \) graph sketches & returning the minimum answer

Each **graph sketch** \( i \) consists of:

- **supernodes**: “node buckets” created by mapping the original nodes via a hash function \( h_i() \)
- **superedges**: sum of the connections between the constituent nodes (of the supernodes they connect)

The more pairwise independent hash functions (sketches)

- => the lower probability of hash collisions
- => the more precise answers to the queries

---

TCM supports conditional node queries, aggregated edge weights, aggregated node flows, reachability path queries, aggregate subgraph queries, triangles

[Tang et al., ‘16]
Influence and diffusion processes are inherently time-evolving. The methods in this category aim at summarizing the influence mainly in social networks.
Summarization of Diffusion Processes in Dynamic Graphs

- **Goal**: interestingness-driven diffusion processes (cascades)
- **Input**:
  - stream of time-ordered interactions, represented as undirected edges between labeled nodes
- **Output**:
  - subgraphs of ‘interesting’ nodes
- **Definition of node interestingness**
  \[ \beta \cdot \log \text{deg}_{out}(v) + (1 - \beta) \cdot \max \text{‘propagation radius’} \]
  path length from the root of the diffusion process to \( v \)

[Qu et al., ‘14]
Main Algorithmic Ideas – **OSNet:**

- construction of spreading trees
- computation of node interestingness
  - nodes are in the summary if interestingness > $\theta$
- interestingness of a summary: min entropy

[Qu et al., '14]
Main Algorithmic Ideas – **OSNet**:

- construction of spreading trees
- computation of node interestingnessness
- interestingness of a summary: min entropy

**VEGAS** [Shi et al. ‘15] also performs summarization by maximizing influence propagation, but only on static graphs

[Qu et al., ‘14]
Summarization of Diffusion Processes in Dynamic Graphs

Diffusion process at $t$ (Zipf)

- OSNet helps understand the dynamics of diffusion processes
- [Toivonen et al. ‘11]: requires user-defined parameters
- [Navlakha et al. ‘09]: finds cliques, which do not help explain diffusion processes

OSNet [Qu et al., ‘14]

[Toivonen et al., ‘11]

[Navlakha et al., ‘09]

[Qu et al., ‘14]
Summarization of Diffusion Processes in Dynamic Graphs

Sample diffusion processes

Sample Summaries

Vocabulary of patterns!

[Qu et al., '14]
Summarization of Social Activity

Understanding collective social activity in over time

NMF on multi-graph (user-photo, user-comment, etc.)

• evolution of themes via cosine similarity

[Lin et al., ‘08]
NetCondense: Motivation

Given temporal graph $G$ find condensed graph $G^\text{cond}$

- merge nodes
- merge time-stamps

Data Mining Tasks

- Influence Maximization
- Community Detection
- Immunization
- Pattern Detection
- Event Detection
- ...

“Preserve” the “propagation based property”

[Adhikari et al., ‘17] – slides adapted with permission
Temporal Network Condensation Problem

Given:
• temporal network $G = \{G_1, G_2, \ldots, G_T\}$
• reduction factors $\alpha_N$ and $\alpha_T$

Find:
• condensed network $G^{cond} = \{G'_1, G'_2, \ldots, G'_{T'}\}$
• Such that $|\lambda_s - \lambda_s^{cond}|$ is minimized

By:
• node- and time-pair merge definitions
1. flatten the given $\mathcal{G}$ to obtain $X_\mathcal{G}$
2. compute $\lambda_x$ and corresponding eigenvector
3. estimate $\Delta$-scores using perturbation
4. sort them in increasing order
5. until the graph is small enough do repeatedly merge best time-pair and node-pairs

Extended to attributed diffusion graphs [Amiri et al. ‘18]
**Problem:** Given a temporal network [Aggarwal +, SDM 2012]
- choose best $k$ nodes in first time-stamp as seed-set
- s.t. maximum diffusion is achieved in the last time-stamp
CONDINF Algorithm

1. Condense the input using NETCONDENSE
2. Solve the temp inf max problem on the condensed network
3. Project the solution back to the original network
   Randomly return a node from a “super-node” is selected

[Adhikari et al., ‘17] – slides adapted with permission
CONDINF finds good answer with significant speed-up

Base method: FI [Aggarwal+, SDM 2012]

CONDINF finds good answer with significant speed-up
Other related work

- **Graph clustering** [Gorke et al. ‘10] [Saha and Mitra ‘07]…
- **Sketches** [Ahn et al. ‘12] [Liberty ‘13]…
- **Compression** [Henecka and Roughan ‘15] [Liu et al. ‘12]…
Summarizing Multiple Disparate Networks

i.e., without time dependencies
Applications of “summaries” of features

**Healthy** and **unhealthy** subjects in neuroscience
- degree
- clustering coefficient
- average path length
- ...

**Anomaly detection** in Twitter
- power laws (degree etc)
- 6-degree of separation
- ...

[Jin, Koutra. IEEE ICDM ’17.]
One summary does not fit all

[Citation: DBPL, Arxiv]

[Social: Twitter, Epinions]

[Neural: Brain, Proteins]

[Jin, Koutra. IEEE ICDM ’17.]
Given: an input graph & domain knowledge

Find: representative features with desired properties (e.g., diversity)

Domain-specific Summarizationization

[PageRank] + [Clust. Coeff.]

Domain knowledge

A collection of graphs with all their features

[Clust. Coeff.]

Graph invariant distributions (PDF)

[Jin, Koutra. IEEE ICDM ’17.]
Other Approaches

Check the centralities!

PageRank
EAGLE: Key Idea

Check the centralities!

PageRank

Clust. Coeff.

Summary

Domain knowledge

[Jin, Koutra. IEEE ICDM ’17.]
Domain-specific Summarization

Requirements for summary:

- diverse
- concise
- domain-specific
- interpretable
- efficient to compute

\[ \text{argmin} \; f \; \text{s.t.} \; \lambda_1 f^T S_F f + \lambda_2 \| f \|_0 + \lambda_3 \varphi(g, G_1, G_2, \ldots, G_K) \]

- diversity
- conciseness
- domain specificity

[Jin, Koutra. IEEE ICDM ’17.]
Although not designed explicitly for this, features selected by EAGLE can be applied to specific tasks, such as classification, with promising performance.

[Jin, Koutra. IEEE ICDM ’17.]
Multiple Networks

- Multi-network summarization is more **challenging** than network-level summarization
  - How to reduce re-computations? pick the right temporal granularity? handle node additions / deletions? make the methods scale to multiple networks?
- **Main focus**: **temporal** networks
  - Applying static methods on snapshots is not sufficient
  - Different models: static snapshots / tensor, graph stream
- **Very limited work** on
  - attributed temporal networks
  - multiple disparate networks
- “One size does not fit all”!
  - we should be thinking about tailored summaries: domain-specific, personalized, query-driven etc.

Big challenges, huge opportunities!
Questions?

Graph Summarization

- Static
  - Plain
  - Labeled
  - Dynamic
    - Plain

Network type

Structure only
- Grouping
- Compression
- Simplification
- Influence

Structure + labels
- Grouping
- Compression
- Influence

Temporal structure
- Grouping
- Compression
- Influence

Core techniques employed

Applications
- Query Efficiency
- Compression
- Visualization
- Pattern Discovery
- Influence Analysis

Open Problems
- Richer Data
- Streaming / Incremental Summarization
- Automated Insight Extraction
- Evaluation

Related Research Areas:
- Graph clustering, partitioning, community detection, sampling, sparsification, sketches, compression
For more details

- Based on survey

https://dl.acm.org/citation.cfm?id=3186727

Graph Summarization Methods and Applications: A Survey

YIKE LIU, TARA SAFAVI, ABHILASH DIGHE, and DANAI KOUTRA, University of Michigan, Ann Arbor

While advances in computing resources have made processing enormous amounts of data possible, human ability to identify patterns in such data has not scaled accordingly. Efficient computational methods for condensing and simplifying data are thus becoming vital for extracting actionable insights. In particular, while data summarization techniques have been studied extensively, only recently has summarizing interconnected data, or graphs, become popular. This survey is a structured, comprehensive overview of the state-of-the-art methods for summarizing graph data. We first broach the motivation behind and the challenges of graph summarization. We then categorize summarization approaches by the type of graphs taken as input and further organize each category by core methodology. Finally, we discuss applications of summarization on real-world graphs and conclude by describing some open problems in the field.

CCS Concepts: • Mathematics of computing → Graph algorithms; • Information systems → Data mining; Summarization; • Human-centered computing → Social network analysis; • Theory of computation → Unsupervised learning and clustering; • Computing methodologies → Network science;

Additional Key Words and Phrases: Graph mining, graph summarization

ACM Reference format:
https://doi.org/10.1145/3186727

1 INTRODUCTION

As technology and access to the amount of data that we can capture and store at our disposal has increased,

49
References


Scalable dynamic graph summarization. Ioanna Tsalouchidou ; Gianmarco De Francisci Morales ; Francesco Bonchi ; Ricardo Baeza-Yates. IEEE International Conference on Big Data (Big Data). 2016


Part III: Local Summarization

Jilles Vreeken