Big Graph Mining: Algorithms, Anomaly Detection, and Applications

U Kang
Korea Advanced Institute of
Science and Technology
ukang@cs.kaist.ac.kr

Leman Akoglu
Stony Brook University
Dept. of Computer Science
leman@cs.stonybrook.edu

Duen Horng (Polo) Chau
Georgia Tech
polo@gatech.edu

ABSTRACT

Graphs are everywhere in our lives: social networks, the World Wide Web, biological networks, and many more. The size of real-world graphs are growing at unprecedented rate, spanning millions and billions of nodes and edges. What are the patterns and anomalies in such massive graphs? How to design scalable algorithms to find them? How can we make sense of very large graphs? And what kind of real-world problems can we solve with such tools? These are exactly the goals of this tutorial.

We start with important graph algorithms that are central to graph mining and pattern discoveries, and describe how we can implement their fast, scalable versions using a unified framework built on top of HADOOP. Then we describe graph-based anomaly detection techniques (complement of pattern discoveries) and how to scale them to massive graphs. Finally, we discuss how our aforementioned techniques can help solve large-scale, real-world problems that make impact to society, and to help solve challenging problems in visual analytics.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms

Design, Experimentation, Algorithms

Keywords

Graph Mining, MapReduce, Hadoop, Visual Analytics, Anomaly Detection

1. INTRODUCTION

Graphs are everywhere: social networks, computer networks, mobile call networks, the World Wide Web [3], protein interaction networks, and many more. The lower cost of disk storage, the success of social networking websites and Web 2.0 applications, and the high availability of data sources lead to graphs being generated at unprecedented size. They are now measured in terabytes or even petabytes, with millions and billions of nodes and edges.

Finding patterns on large graphs have a lot of applications including cyber security [22], social network analysis (Facebook, Twitter) [16], and fraud detection [8], among others. This tutorial addresses the problem of *finding patterns and anomalies in large-scale graphs with massively scalable algorithms and tools.* Specifically, we aim to answer the following questions: How can we scale up graph mining algorithms for massive graphs with billions of edges? How can we find anomalies in such large-scale graphs? How can we make sense of disk-resident large graphs, what and how can we do visual analytics? How can we use the algorithms and anomaly detection techniques to solve challenging real-world problems?

Our tutorial consists of three main parts. We start with scalable algorithms for large graph mining for billion-scale graphs, including structure analysis, eigensolvers, storage and indexing, and graph layout and graph compression. Next we describe anomaly detection techniques for large scale graphs. Finally, we discuss the applications and visual analytics which leverage these algorithms and anomaly detection techniques in the previous parts.

Length.

Our tutorial is planned to take 3 hours: 1 hour for algorithms, 1 hour for anomaly detection, and 1 hour for applications and visualization, which we describe in detail in Section 2.

Target Audience.

The target audience consists of social network, data bases, and data mining professionals who wish to have a comprehensive understandings on large-scale graph mining and management algorithms, applications, and anomaly detection tools. The audience will learn recent developments on big graph mining and how they could utilize these tools for real-world problems that they are facing with in the wild.

Prerequisites. Computer science background (B.Sc. or equivalent); familiarity with undergraduate linear algebra.

Relation to Previous Tutorials by the Authors.

We list the similarities and differences between this tutorial and the authors' previous tutorials.

- A portion of the *algorithms* part (Section 2.1) of this

ASONAM'13, August 25–28, 2013, Niagara Falls, Canada.

tutorial appeared in SIGMOD'12 tutorial "Mining Billion-Scale Graphs: Patterns and Algorithms"[1]. We extend it to include graph storage and indexing techniques in very large graphs with billions of nodes and edges. The main novelties in this tutorial are emphasis on anomaly detection, as well as recent developments in visual analytics for large graphs.

- Akoglu has taught two related tutorials on graph anomaly detection (Section 2.2): "What is Strange in Large Networks? Graph-based Irregularity and Fraud Detection" (ICDM'12, 4hrs)[2], and "Anomaly, Event, and Fraud Detection in Large Graph Datasets" (WSDM'13, 6hrs)[3]. Different from those, this tutorial will focus on the scalability aspects and challenges of graph anomaly detection techniques for massive scale graphs.

Relation to Previous Tutorials by Other People.
There have been tutorials on graph mining and anomaly detection in general, not discussing the problems and techniques one is confronted with when mining massive, tera- to peta-scale graphs. Our main contribution we aim to introduce with this tutorial is the scalability aspects of graph mining algorithms and tools.

- *Anomaly Detection: A Tutorial*, [ICDM 2011] by S. Chawla and V. Chandola.
- *Outlier Detection Techniques*, [KDD 2010] by H.-P. Kriegel, P. KrÃűger, and A. Zimek.
- *Anomaly Detection*, [SDM 2008] by A. Banerjee, V. Chandola, V. Kumar, and J. Srivastava.
- *Data Mining for Anomaly Detection*, [ECML PKDD 2008] by A. Lazarevic, J. Srivastava, V. Kumar, A. Banerjee, and V. Chandola.
- *Graph mining*: ICDE'09, CIKM'08, KDD'04

The main focus of our tutorial is scalability, i.e. notably different from, while being complementary to these earlier tutorials. More specifically,

1. Unlike the previous tutorials on graph mining, we focus on scalable graph mining, and cover a comprehensive list of techniques to scale graph mining algorithms to disk-resident massive-scale graphs.

2. Unlike the previous tutorials on anomaly detection, we focus specifically on finding abnormalities in (large) graph datasets.

3. In addition, we focus on large-scale graph visualization and sensemaking, and highlight the real-world applications of our presented tools and algorithms in the wild, including fraud and malware detection.

Specific Audio/Video/Computer Requirements.
There is no specific requirement.

2. TUTORIAL OUTLINE

Our tutorial consists of three inter-related parts.

- *Scalable Algorithms*, focusing on large graph mining on Hadoop, including structure analysis, eigensolver, storage/indexing, and graph layout/compression;
- *Anomaly Detection*, introducing anomaly detection techniques in graph datasets as well as graph algorithms that help with anomaly detection in relational data;
- *Applications and Visualization*, leveraging algorithms and anomaly detection techniques mentioned in the previous parts for real-world applications including fraud, spam, malware, etc. detection, and sensemaking and analysis of billion-node graphs.

In what follows, we describe the goals and deliverables of each part in detail.

2.1 Scalable Algorithms for Large-scale Graph Mining

How to scale up the algorithms for mining very large graphs which do not fit in memory, or disks of a single machine? How to use parallelism? We describe how to design and implement such algorithms on Hadoop. The algorithms are general and cover diverse cases including the structural analysis, eigensolver, storage/indexing, and compression.

Structure Analysis. How can we find connected components, diameter, PageRank, and node proximities of very large graphs quickly? Furthermore, how can we design a general primitive which can be applied to many different algorithms? We describe `GIM-V` (`G`eneralized `I`terative `M`atrix-`V`ector multiplication) [21], an important primitive which unifies many seemingly different algorithms including connected components, diameter [19, 20], PageRank, and node proximities. We also describe how to develop fast algorithms for `GIM-V` on MapReduce framework.

Eigensolver. Given a billion-scale graph, how can we find near-cliques, the count of triangles, and related graph properties? All of them can be found quickly if we have the first several eigenvalues and eigenvectors of the adjacency matrix of the graph [34, 30]. Despite their importance, however, existing eigensolvers do not scale well. We introduce HEigen [16], an eigensolver for billion-scale, sparse matrices. We describe the design decisions and fast algorithms that enable the scalable billion-scale eigensolver.

Storage and Indexing. How to store and index graph edge files so that graph mining queries can be answered quickly? Graph storage and indexing are important especially for targeted graph mining queries whose answers require the access to only parts of the graph. Examples of targeted queries include k-step in/out-neighbors, and egonet queries. We describe how to store and index the nonzero elements in the adjacency matrix of the graphs to quickly answer graph mining queries [17].

Graph Layout and Compression. Given a real world graph, how should we lay-out its edges? How can we compress it? These questions are closely related, and the typical approach so far is to find clique-like communities, like the 'cavemen graph', and compress them. We show that the block-diagonal mental image of the 'cavemen graph' is the wrong paradigm, in full agreement with earlier results that real world graphs have no good cuts. We describe the recent development for graph compression, called SlashBurn

[1] `http://www.cs.cmu.edu/~christos/TALKS/12-SIGMOD-tutorial/`
[2] `http://www.cs.stonybrook.edu/~leman/icdm12/`
[3] `http://www.cs.stonybrook.edu/~leman/wsdm13/`

method [14] which has several advantages: (a) it avoids the 'no good cuts' problem, (b) it gives better compression, and (c) it leads to faster execution times for matrix-vector operations, which are the back-bone of most graph processing tools [18, 21].

2.2 Anomaly Detection for Large-scale Graph Mining

The second part of the tutorial will delve into the 'detecting rare occurrences in large graphs' problem. This problem domain has numerous applications in security, finance, health care, law enforcement, etc. such as detecting network intrusion or network failure, credit card fraud, insurance claim fraud, accounting fraud, email, Web, or opinion spam, auction fraud, and many others.

Different from earlier tutorials on the outlier/anomaly detection topic, we will focus on the problem domain for graph (or network, relational) data. Data objects interlinked by edges in a graph representation have long-range correlations, and hence the problem of finding patterns and anomalies in graph datasets require novel technology, with the problem becoming more challenging with large-scale and dynamically changing graphs.

The main goal of this third part of the tutorial is to provide a comprehensive and unified overview of the state-of-the-art techniques for anomaly detection in massive data represented as graphs. The proposed contributions/parts of this part are the following.

1. *Graph-based anomaly detection at large-scale:* We give a comprehensive overview of abnormality detection techniques for graph data, and discuss how to scale these techniques to massive graphs (this is in complement with the first part of the tutorial). Relevant references include [32, 12, 26, 9, 2, 33].
2. *Advantages/Challenges:* We thoroughly explore techniques from both data mining (unsupervised, exploiting graph structure) and machine learning ((semi-) supervised, employing relational learning), and discuss their pros/cons in face of big data. Relevant references include [28, 31, 11, 15, 24]

2.3 Applications and Visualizations for Large-scale Graph Mining

The last part of the tutorial discusses how the scalable graph algorithms and anomaly detection techniques presented in our previous parts can help solve large-scale, real world problems that make impact to society, and to help solve challenging problems in visual analytics.

High-Impact Applications. We will discuss how scalable algorithms such as Belief Propagation (BP) [13] (in Symantec's *Polonium* technology [5]) is helping to protect over 120 million people worldwide from malware, by inferring every file's reputation, flagging files with low reputation as malware; the algorithm works over a massive graph that describes over 37 billion machine-file relationships. We will also discuss how variations of BP have been used in other applications, such as spotting auction fraud (in *NetProbe* [27]) and accounting fraud (in *Snare* [23]).

We will discuss how to leverage eigenvectors computed by scalable eigensolver to spot anomalies in graphs such as *near-cliques* (e.g., close-knit groups in mobile call graph) and *near-bipartite* cores (e.g., reusing citations by patents filed by the same company). Pairs of eigenvectors form the striking EigenSpokes pattern [29] which, when plotted against each other, form highly separate lines. Other applications we will cover includes using graph mining to spot security fraud [25], opinion spam [35], web spam [10], and more.

Visual Analytics. Traditionally, visualization systems rely heavily on the user manually picking what to visualize or analyze. But for large graphs, finding a good starting point to investigate becomes a difficult task, as users can no longer visually distill points of interest. We will present a recent idea called *Attention Routing* [4] that aims to overcome this critical problem in visual analytics, which leverages graph-based anomaly techniques (e.g., OddBall [2]) to channel users' attention through massive networks to interesting nodes or subgraphs which may represent good starting points for analysis.

Merely locating good starting points is not enough, however. It is often important to help users understand the context surrounding those starting points (e.g., who are the neighbors?) However, this presents serious problems for massive graphs: a node in such graphs can easily have thousands of neighbors. Where should the user go next? Which neighbors to show? We will present several interactive, mixed-initiative systems that combine scalable graph mining techniques with intuitive user interfaces to help people explore large graphs, such as 1) the *Apolo* system [7] which incorporate users' feedback, as to which nodes are relevant nodes, and use that to recommend which other areas the user may want to see next; 2) the *Graphite* system [6] that finds both exact and approximate matches for user-specified subgraphs, which is particularly useful when the user has some idea about the kind of subgraphs to look for, but is not able to describe it precisely; 3) the *OPAvion* system [1] which provides an interactive user interface for the user to interact with a distributed computation module that does the heavy lifting of graph mining.

3. ABOUT THE INSTRUCTORS

U Kang is an assistant professor in Computer Science Department at KAIST. He won two best paper awards, and he has published many refereed articles in major data mining and database venues. He holds two U.S. patents. His research interests include data mining in big graphs. He leads the research of the award-winning PEGASUS software [21].

Leman Akoglu is an assistant professor at Stony Brook University. She received her Ph.D. from Carnegie Mellon University in 2012. Her research has won 2 "Best Paper" awards, and led to several peer-reviewed publications in major data mining venues, and 3 U.S. patents, filed by IBM T. J. Watson Research. Her research interests are in data mining, machine learning, and applied statistics with a focus on pattern mining, and anomaly and event detection in large dynamic data using graph mining and compression.

Duen Horng (Polo) Chau is an Assistant Professor at Georgia Tech's College of Computing. His Ph.D. thesis received Carnegie Mellon's Computer Science Dissertation Award, runner up for bridging Data Mining and Human-Computer Interaction (HCI) for making sense of large graphs. His Polonium malware detection technology (with Symantec, patented) helps protect 120 million people worldwide. Polo is the only two-time Symantec fellow. His NetProbe auction fraud detection system appeared on The Wall Street Journal, CNN, TV and radio.

4. REFERENCES

[1] L. Akoglu, D. H. Chau, U. Kang, D. Koutra, and C. Faloutsos. Opavion: Mining and visualization in large graphs. In *SIGMOD*, pages 717–720. ACM, 2012.

[2] L. Akoglu, M. McGlohon, and C. Faloutsos. OddBall: Spotting anomalies in weighted graphs. In *PAKDD*, 2010.

[3] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer Networks 33*, 2000.

[4] D. Chau. *Data Mining Meets HCI*. PhD thesis, Carnegie Mellon University, 2012.

[5] D. Chau, C. Nachenberg, J. Willhelm, A. Wright, and C. Faloutsos. Polonium: Tera-scale graph mining and inference for malware detection. In *SIAM SDM*, 2011.

[6] D. H. Chau, C. Faloutsos, H. Tong, J. I. Hong, B. Gallagher, and T. Eliassi-Rad. Graphite: A visual query system for large graphs. In *Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on*, pages 963–966. IEEE, 2008.

[7] D. H. Chau, A. Kittur, J. I. Hong, and C. Faloutsos. Apolo: making sense of large network data by combining rich user interaction and machine learning. In *Proceedings of the 2011 annual conference on human factors in computing systems*, pages 167–176. ACM, 2011.

[8] D. H. Chau, S. Pandit, and C. Faloutsos. Detecting fraudulent personalities in networks of online auctioneers. *PKDD*, 2006.

[9] W. Eberle and L. B. Holder. Anomaly detection in data represented as graphs. *Intell. Data Anal.*, 11(6):663–689, 2007.

[10] Z. Gyogyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *Proc. VLDB*, 2004.

[11] K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, and C. Faloutsos. It's who you know: Graph mining using recursive structural features. In *KDD*, pages 663–671, 2011.

[12] T. Ide and H. Kashima. Eigenspace-based anomaly detection in computer systems. In *KDD*, pages 440–449, 2004.

[13] U. Kang, D. H. Chau, and C. Faloutsos. Mining large graphs: Algorithms, inference, and discoveries. In *ICDE*, pages 243–254. IEEE, 2011.

[14] U. Kang and C. Faloutsos. Beyond 'caveman communities': Hubs and spokes for graph compression and mining. In *ICDM*, 2011.

[15] U. Kang, M. McGlohon, L. Akoglu, and C. Faloutsos. Patterns on the connected components of terabyte-scale graphs. In *ICDM*, pages 875–880, 2010.

[16] U. Kang, B. Meeder, and C. Faloutsos. Spectral analysis for billion-scale graphs: Discoveries and implementation. In *PAKDD (2)*, pages 13–25, 2011.

[17] U. Kang, H. Tong, J. Sun, C.-Y. Lin, and C. Faloutsos. Gbase: a scalable and general graph management system. In *KDD*, pages 1091–1099, 2011.

[18] U. Kang, C. Tsourakakis, and C. Faloutsos. Pegasus: A peta-scale graph mining system - implementation and observations. *ICDM*, 2009.

[19] U. Kang, C. E. Tsourakakis, A. P. Appel, C. Faloutsos, and J. Leskovec. Radius plots for mining tera-byte scale graphs: Algorithms, patterns, and observations. In *SDM*, pages 548–558, 2010.

[20] U. Kang, C. E. Tsourakakis, A. P. Appel, C. Faloutsos, and J. Leskovec. Hadi: Mining radii of large graphs. *ACM Trans. Knowl. Discov. Data*, 5:8:1–8:24, February 2011.

[21] U. Kang, C. E. Tsourakakis, and C. Faloutsos. Pegasus: mining peta-scale graphs. *Knowl. Inf. Syst.*, 27(2):303–325, 2011.

[22] K. Maruhashi, F. Guo, and C. Faloutsos. Multiaspectforensics: Pattern mining on large-scale heterogeneous networks with tensor analysis. In *ASONAM*, pages 203–210, 2011.

[23] M. McGlohon, S. Bay, M. G. Anderle, D. M. Steier, and C. Faloutsos. Snare: a link analytic system for graph labeling and risk detection. In *KDD*, pages 1265–1274, 2009.

[24] J. Neville and D. Jensen. Collective classification with relational dependency networks. *Journal of Machine Learning Research*, 8:2007, 2003.

[25] J. Neville, O. Simsek, D. Jensen, J. Komoroske, K. Palmer, and H. G. Goldberg. Using relational knowledge discovery to prevent securities fraud. In *KDD*, pages 449–458, 2005.

[26] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *KDD*, pages 631–636, 2003.

[27] S. Pandit, D. H. Chau, S. Wang, and C. Faloutsos. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *WWW*, 2007.

[28] B. Pincombe. Anomaly detection in time series of graphs using arma processes. *ASOR Bulletin.*, 24(4):2–10, 2005.

[29] B. Prakash, A. Sridharan, M. Seshadri, S. Machiraju, and C. Faloutsos. Eigenspokes: Surprising patterns and scalable community chipping in large graphs. *Advances in Knowledge Discovery and Data Mining*, pages 435–448, 2010.

[30] B. A. Prakash, M. Seshadri, A. Sridharan, S. Machiraju, and C. Faloutsos. Eigenspokes: Surprising patterns and community structure in large graphs. *PAKDD*, 2010.

[31] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[32] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *KDD*, pages 687–696, 2007.

[33] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *ICDM*, pages 418–425, 2005.

[34] C. Tsourakakis. Fast counting of triangles in large real networks without counting: Algorithms and laws. In *ICDM*, 2008.

[35] G. Wang, S. Xie, B. L. 0001, and P. S. Yu. Review graph based online store review spammer detection. In *ICDM*, pages 1242–1247, 2011.