Current and Future Challenges in Mining Large Networks: Report on the Second SDM Workshop on Mining Networks and Graphs

Lawrence B. Holder
Washington State University

Maleq Khan
Virginia Tech

Christine Klymko
Lawrence Livermore

Rajmonda Caceres
MIT Lincoln Lab

Nitesh V. Chawla
University of Notre Dame

Tina Eliassi-Rad
Rutgers University

David F. Gleich
Purdue University

Ravi Kumar
Google, Inc.

Aditya Prakash
Virginia Tech

Jason Riedy
Georgia Tech

Yinghui Wu
Washington State University

ABSTRACT

We report on the Second Workshop on Mining Networks and Graphs held at the 2015 SIAM International Conference on Data Mining. This half-day workshop consisted of a keynote talk, four technical paper presentations, one demonstration, and a panel on future challenges in mining large networks. We summarize the main highlights of the workshop, including expanded written summaries of the future challenges provided by the panelists. The current and future challenges discussed at the workshop and elaborated here provide valuable guidance for future research in the field.

Keywords

Network mining, graph mining, big data, challenges.

1. INTRODUCTION

Real-world applications give rise to networks that are unstructured and often comprised of several components. Furthermore, they can support multiple dynamical processes that shape the network over time. Network science refers to the broad discipline that seeks to understand the underlying principles that govern the synthesis, analysis and co-evolution of networks. In some cases, the data relevant for mining patterns and making decisions comes from multiple heterogeneous sources and streams in over time. Graphs are a popular representation for such data because of their ability to represent different entity and relationship types, including the temporal relationships necessary to represent the dynamics of a data stream. However, fusing such heterogeneous data into a single graph or multiple related graphs and mining them are challenging tasks. Emerging massive data has made such tasks even more challenging.

The 2015 SDM Workshop on Mining Networks and Graphs [21] brought together researchers and practitioners in the field to deal with the emerging challenges in processing and mining large-scale networks. Such networks can be directed as well as undirected, they can be labeled or unlabeled, weighted or unweighted, and static or dynamic. Networks of networks are also of interest. Specific scientific topics of interest for this meeting include mining for patterns of interest in networks, efficient algorithms (sequential/parallel, exact/approximation) for analyzing network properties, methods for processing large networks (i.e., Map-Reduce and Giraph based frameworks), use of linear algebra and numerical analysis for mining complex networks, database techniques for processing networks, and fusion of heterogeneous data sources into graphs. Another particular topic of interest is to couple structural properties of networks to the dynamics over networks, e.g., contagions.

The workshop consisted of a keynote talk by Ravi Kumar from Google, four technical paper presentations, a demonstration of the CINET Cyberinfrastructure for Network Science by Maleq Khan from Virginia Tech, and a panel on Future Challenges in Mining Large Networks. The panelists included Rajmonda Caceres from MIT Lincoln Lab, Nitesh Chawla from Notre Dame, Tina Eliassi-Rad from Rutgers, David Gleich from Purdue, Christine Klymko from Lawrence Livermore, Ravi Kumar from Google, Jason Riedy from Georgia Tech, Aditya Prakash from Virginia Tech, and Yinghui Wu[1] from Washington State. The workshop was co-chaired by Lawrence Holder from Washington State, Maleq Khan from Virginia Tech, and Christine Klymko.

In the following sections we summarize the presentations and discussions at the workshop. Each panelist has also provided a written summary elaborating on their future challenge.

2. CURRENT DIRECTIONS FOR MINING NETWORKS AND GRAPHS

Ravi Kumar gave a keynote talk entitled "Estimating Network Parameters." Estimating the parameters such as the size and average degree of a large network, which cannot be accessed in its entirety, is a basic data mining question. Recently, the problems of estimating the size of the web, the size of a web index, the size and other parameters of online social networks, etc. have been actively considered in the context of World Wide Web [12]. In this talk, Ravi Kumar addressed several questions with the main focus on estimating the network size and the average degree. The main motivation of estimating these parameters is to understand the network in general. In the case of social network, it can help in gaining business insight and competitive advantage [12]. These

[1] Yinghui Wu was unable to attend the workshop due to last minute visa issues, but we have included the written summary of his challenge in this report.

problems become challenging and interesting with the following realistic assumptions: i) the network is not available to us in its entirety – we can only query a node and obtain all its neighbors, ii) these queries are expensive, and thus an algorithm has to make a small number of queries, and iii) it may not be possible to access a uniformly random node in the network. The speaker discussed some traditional methods and then showed some recently developed advanced techniques that reduce the number of queries significantly.

Four contributed papers [2, 10, 16, 38] were presented in the workshop. These papers have also been published in the workshop proceedings. In [2], the authors addressed the problem of mining coevolving patterns in dynamic networks. They present an algorithm to analyze all relational changes between entities (nodes) and find all frequent coevolving induced relational motifs. Their results show that these motifs capture network characteristics that can be useful for modeling the underlying dynamic network. A recent trend and important problem in graph mining is to mine social, financial, or other relevant networks for detecting intrusion and suspicious activities. Another paper [16] presents a method of detecting intrusion using frequent subgraphs. Community detection in a network is another important problem and recently received significant attention of the researchers. Large-scale networks (networks with billions of nodes and edges) require very efficient algorithms. Some efficient methods for detecting communities in large-scale networks are presented in [38] and [10].

Maleq Khan gave a demonstration of an open–access web-based network analysis tool called CINET [1, 15], a Cyber Infrastructure for NETwork Science[2]. CINET has been developed at Virginia Tech and partially funded by NSF. It provides a large set of networks and modules (such as computing diameter, clustering coefficient and shortest path) to analyze them. Users can also add their own networks to be analyzed by the provided algorithms. The web-based interface has been designed to simplify analysis of complex networks for users who are not necessarily computer scientists.

3. FUTURE CHALLENGES ON MINING LARGE NETWORKS

While the panelists had only three minutes each to present their challenge at the workshop, they have also provided written descriptions after the workshop, which are included here.

3.1 Graph Representation Learning[3]

Rajmonda Caceres, MIT Lincoln Laboratory

The process of going from raw data to the right graph representation is a critical building block for a successful data-to-decisions analytical framework. When properly done, the graph representation captures the essential aspects of the data and abstracts away the noisy, irrelevant parts. Many inference algorithms make two fundamental assumptions: 1) the graph is already constructed 2) the constructed graph has the qualitative properties necessary for their analysis to work, i.e., the patterns that we are looking for are present and recoverable. In reality, what we have available is raw data that is often noisy and collected from different modalities. Furthermore, no clear methodology exists in place for converting these data into a useful graph representation. Current practices often aggregate different graph sources ad-hoc, making it difficult to compare algorithms across different domains or even within the same domain using different data sources. The immediacy for rigorous approaches on representation learning of graphs is even more apparent in the big data regime, where challenges connected to variety and veracity exacerbate the challenges of volume and velocity.

Constructing quality graph representations from raw data is a challenging task. Often the data we collect represent indirect measurement of the true relationships we want to analyze, for example, we want to analyze social relationships, but we collect proximity information. Data collections systems often introduce a lot of noise in the form of missing or irrelevant connections. Finally, it is not clear how to integrate different, potentially complementary data sources into one unified representation.

An orthogonal challenge has to do with our mathematical understanding (or lack of) of what makes a graph representation qualitative. If we did have a good understanding of this, we could then hope to design algorithms to drive the data-to-graph mapping in the right direction. In reality, we do not have ground truth, nor do we have notions of quality that we agree upon. More importantly, we often observe that the quality of graph representation depends on the objective of the learning task, and for the same learning task, multiple graph representations might be useful.

A much-needed capability in this problem setting is one that takes multi-source, incomplete, noisy data and constructs quality networks together with estimations of uncertainty/confidence of the network components (edges, subgraphs, etc.). There are additional related open research questions and potential areas of impact, from developing methods for validating the quality of graph representation in the absence of ground truth, to identifying scenarios when fusion of different sources helps, to deriving performance guarantees for different graph construction or graph recovery techniques.

3.2 Representing Higher-Order Dependencies in Networks

Nitesh V. Chawla, University of Notre Dame

How to construct the network representation from data, such that the underlying phenomena in data are correctly captured and represented?

The conventional way of constructing a network from raw data typically assumes the Markov property (first order dependency) by considering only the pairwise connections in data. That is, in such a network, a movement simulation (such as trajectories of vehicles, retweets, clickstream traffic, etc.) is only able to follow the probability distribution of the first order, and cannot reflect higher order dependencies that may exist in the data. This can lead to inaccuracies when applying a wide range of network analyses tools that are based on the simulations of movements in the network, such as clustering, PageRank, various link prediction methods based on random walking, and so on.

[2] http://www.vbi.vt.edu/ndssl/cinet.

[3] This challenge is part of work sponsored by the Department of the Air Force under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Specifically, the challenge that we posit is as follows. Construct a network that accurately captures the variable and higher order of dependencies such that it is:

a) representative of the underlying phenomena in the data to more accurately represent simulation of movement

b) compact in size allowing for variable order of dependencies versus using a fixed high order

c) compatible with existing network analysis tools such that the analysis toolkit does not have to change to respond to the network representation

3.3 Provably Increasing Network Awareness

Tina Eliassi-Rad, Rutgers University

The underlying processes generating network data are often partially observed. Thus, regardless of how big the data is, it is incomplete and noisy. For example, current maps of the Internet are known to be incomplete and significantly biased [18]. The challenge is to provably increase network awareness. Specifically, given an incomplete, noisy, and possibly biased network, can we infer network properties (at micro, mezzo, and macro levels) with provable accuracy? Then, given these inferences can we design active graph probing/learning algorithms for graph mining tasks (such as community detection, role extraction, etc.)? Approaches from computer science theory such as property testing [14, 32] and sublinear algorithms [33] and from machine learning such as active learning [6, 30] are possible solutions to this solution.

This challenge is joint work with Sucheta Soundarajan (Rutgers University), Brian Gallagher (Lawrence Livermore National Laboratory), Ali Pinar (Sandia National Laboratories), C. Seshadhri (University of California Santa Cruz), and Bradley Huffaker (CAIDA).

3.4 A Turing Test for Synthetic Network Models

David F. Gleich, Purdue University

We propose establishing a Turing-like test to assess the current state of synthetic network models. Synthetic network models are important for two problems: (i) assessing the statistical significance of results in networks [28] and (ii) measuring the performance of new algorithms on extremely large graphs [7]. But there is widespread disagreement about the relevance of the current state of synthetic generators. New models are constantly being proposed to fit the latest observed feature of real-world networks (see, for instance [24]). The ones that see widespread use are often due to reasons that are distinct from their accuracy as far as modeling real networks [29, 36]. The basis for our proposal is to quantify the current state of synthetic network models and address the question: can we distinguish the distribution of graphs generated by synthetic methods from the distribution of graphs that are from the real-world?

A hypothetical model for such a test is as follows. At the start of each month, there is a new collection of networks released. These networks are either generated by a synthetic generator or a piece of a real-world dataset. At the end of the month, challengers would submit their results on if they believe that the network was the result of a generator or a real-world network. If the current state of synthetic network generation is sufficient, then the two distributions should be indistinguishable. If there are distinguishing features, this suggests how we need to improve current synthetic generators.

Justification. One of the irksome questions in graph mining is trying to determine if a finding is significant or if should have been expected given the known properties of social networks. An approach to answer this question for many subgraph and subset queries involves studying synthetic network models of networks and evaluating the likelihood of finding that subgraph or subset in the synthetic model (or one with similar properties). But this methodology is only useful if the synthetic model *has* the properties that are known be associated with the original class of networks and also has some variance over the distribution of graphs [27]. It is unclear if the current class of synthetic models meets these requirements and the Turing test proposed above would help us answer that question and would also suggest important properties to distinguish real-world networks from their synthetic approximations.

Additionally, extremely large graphs are difficult to find outside of a small number of select institutions such as Google and the NSA. The largest publically available network is 126B edges and 6B vertices (http://webdatacommons.org). There are many problems with this graph that can be solved on a modern laptop computer [26]. One of the approaches to overcome the lack of data is to evaluate synthetic networks that can be generated at arbitrary size-scales. But the relevance of these networks to algorithmic performance is questionable if the underlying networks are not a reasonable approximation. This is especially important for things like partitioning problems where many synthetic networks have relatively simple optimal partitioning strategies.

3.5 Noisy Data and Fuzzy Subgraph Detection

Christine Klymko, Lawrence Livermore National Laboratory

One important issue in dealing with network data is how to account for noise. Noisy data can result from a variety of processes, including: collection error (missing edges, false edges, etc.), mutations (such as those occurring in certain biological networks), actual but unimportant/meaningless interactions (i.e., wrong number phone calls), and nodes attempting to hide their interactions in a network (such as might occur in various social or cybersecurity applications). The presence of noise complicates many data mining problems: see [17, 37], among others.

An example of the difficulties of data mining tasks in the presence of noisy data is the question of subgraph/network motif detection, which becomes especially complicated when noise is taken into account. Subgraph detection is important in a number of areas [19, 25]. However, given the presence of noise, it does not make sense to search for exact subgraphs. Instead, a search for "fuzzy" subgraphs (allowing the addition or deletion of a small number of nodes and edges to the original search query) will often produce more meaningful results. However, there are still few methodologies to effectively perform fuzzy subgraph detection. The development of noise robust methodologies (for subgraph detection and other data mining questions) is an important area of research.

3.6 Scalable Graph Algorithms in Emerging Computational Models

Ravi Kumar, Google

The challenge is to develop and study computational models that are best suited for large data, especially, large graphs. Modern computing paradigms such as streaming and map-reduce have been very useful in developing algorithms that can scale to large

data; these paradigms are reasonably well-established by now and their limitations are well understood. Emerging models such as the asynchronous computational model and the parameter-server model (popular in the machine learning community) seem promising for many new classes of problems; their power and limitations are yet to be understood both from theoretical and applied points of view. It becomes important to study these models and see their applicability to large-scale graph problems – the topic is nascent and rich.

3.7 Error and Sensitivity Analysis for Graphs

Jason Riedy, Georgia Institute of Technology

Most current graph analysis methods assume correct data and knowledge. However, this rarely occurs. We have little knowledge about and fewer models of the sensitivity of analysis results to errors. Graphs imperfectly represent some real phenomenon. "Friendships" in online social networks do not always reflect personal relationships, or the data is obscured for privacy reasons as in health data. Computation imperfectly analyzes the graph. Many problems are only approximated to fit within time or energy limitations. Many codes have subtle bugs. If some problem occurs once in a billion edges, massive graphs will uncover it. Other scientific computing areas have established frameworks for analyzing and addressing sensitivity to perturbations. We need mental and formal methods for addressing error and sensitivity in graph analysis results, and we need to condense those into rules of thumb for practitioners.

The wide range of graph analysis tasks will need a variety of approaches. Globally averaged properties like a graph's clustering coefficient often are not very sensitive to perturbations. Local properties, however, can be affected drastically. Experiments in Zakrzewska and Bader [40] imply that for a variety of graphs and edge dropping heuristics, nearly a quarter of the edges could be ignored while affecting the global clustering coefficient by at most 10%. The vector of local clustering coefficients changes in one-norm relative difference by 20% to 80% in the same range. Consider measuring or modeling error in connected components. The interpretation of error will change depending on the source of the graph data. If the graph is derived from thresholds, say from significance of protein-protein interaction measurements [4], the single threshold may provide leverage in defining a model for the overall graph. Discrete interaction networks as occur in criminal network analysis [8] will require other prediction methods, although meaningfully predicting interactions between disconnected components is (to this author's knowledge) an open problem.

Understanding graph analysis algorithms' sensitivity to error and perturbation is a step towards making graph analysis a solid scientific computing approach. Other scientific computing disciplines have error analysis frameworks that are distilled into basic rules of thumb for practitioners. We need to provide analysts and scientists with the same level of support for confidence in the graph analysis results.

3.8 Propagation over Networks

Aditya Prakash, Virginia Tech

How do contagions like Ebola and Influenza spread in population networks? How do malware propagate? How can blackouts spread on a nationwide scale? How do rumors spread on Twitter/Facebook? Which group should we market to for maximizing product penetration? Answering all these big-data questions involves the study of aggregated dynamics over complex connectivity patterns [5, 20, 23, 31]. Dynamic processes over networks can give rise to fascinating macroscopic behavior, leading to fundamental research problems which recur in multiple domains. Understanding such propagation processes will eventually enable us to manipulate them for our benefit, e.g., understanding dynamics of epidemic spreading over graphs helps design more robust policies for immunization.

These problems are typically very challenging, as they involve high-impact real-world applications as well as deep technical issues like the need for scalability and handling of heterogeneous noisy data in a principled manner. Data for these problems will typically come from domains like epidemiology and public health (both simulation and real data), social media (tweets, blog posts, movie ratings), cyber security (malware databases), historical (newspapers) and so on. Moreover, promising approaches seem to be very inter-disciplinary – drawing concepts and techniques ranging from theory and algorithms (combinatorial and stochastic optimization), systems (asynchronous computation) to machine learning/statistics (minimum description length, graphical models) and non-linear dynamics. Clearly, progress in this sphere holds great scientific as well as commercial value.

3.9 Resource-bounded Graph Mining

Yinghui Wu, Washington State University

An emerging challenge is to develop scalable mining techniques over massive network data with limited resource. Graph mining tasks such as subgraph pattern discovery are inherently expensive, and it is often hard to theoretically reduce the complexity. On the other hand, emerging applications require mining with limited computing resource, such as response time, space cost, energy constraints, etc. For example, applications in cyber network monitoring typically require the anomaly communication patterns be discovered in real-time [11]. The need for big graph mining with bounded resource and (guaranteed) high accuracy is evident in resource-intensive applications.

Recent study on resource bounded and budgeted graph search suggests to explore bounded fraction of graph data to generate approximate answers [13]. Data sketch, summary and compression techniques are applied to generate and query small synopsis from original graphs [3]. The effectiveness and possible performance guarantees of these approaches may rely on specific query classes, domain knowledge and data properties. A possible future direction is to leverage learning techniques and design resource-accuracy trade-off mining algorithms upon specific application need. This may also lead to adaptive mining tools that support large-scale graph analytics in cloud services.

3.10 Panel Discussion

In summary, the presented challenges focused on how best to represent data as a graph, especially noisy data with higher-order dependencies, and how to evaluate the quality of the resulting graph. Since any constructed graph necessarily represents a sample of the real world, how can we assess the quality of the sample and the certainty of the conclusions drawn from the data (e.g., error, sensitivity, and p-value for graphs)? Addressing these issues will help with other challenges related to the design and testing of scalable graph mining algorithms that take maximum advantage of limited resources. After the panelists presented their challenges, a lively discussion ensued among the panelists as they responded to questions from the audience and amongst themselves. Here, we summarize this discussion.

An interesting comment by one of the panelists related the experience of *how seemingly deterministic graph algorithms may yield different results simply by relabeling the nodes* in the graph. A question from the audience asked for an elaboration of the reasons behind such behavior, and the main reasons were the arbitrary ranking among nodes with equivalent values and the precision errors when computing these values, which may be extremely small or large. One panelist asked if this was really a problem, given that we do not always need exact answers to graph problems, e.g., when merely ranking nodes. Others pointed out that if these error-tolerant tasks are repeated or are part of a larger workflow, then errors may propagate, which brings us back to one of the focuses of the challenges: how to assess error in the networks and in the results of graph algorithms. In general, graph analysis is often interested in the solution and not necessarily in optimizing a specific metric. Approaching such a highly non-linear and bumpy problem from different directions/permutations will likely result in different locally-optimal solutions. This is a challenge as it expands the space of viable solutions and complicates the evaluation of algorithms for mining large networks.

Next, one of the panelists proposed a straw man argument of *whether truly big real-world graphs exist, or at least graphs whose size requires more memory and computational power than a modern laptop*. More realistically, are there large graphs that exceed readily available computational resources that cost less than $10,000? Specifically, while Facebook purports to have a real-world graph on the order of one trillion edges [34], and the National Security Agency purports to have a real-world graph with 70 trillion edges requiring one petabyte of storage [7], the largest publicly-available graph has around 128 billion edges [39]. The panelist argued that for most graph mining tasks, a laptop is sufficient for processing a graph on the order of 100 billion edges. Other panelists pointed out that even larger graphs can be constructed by combining multi-typed data from different sources (e.g., all of the web), or incorporating time as in clickstream and network traffic flow data. While such graphs are typically sampled from, filtered, or abstracted in order to fit within memory requirements, simply loading these graphs into memory can take hours. And computationally complex algorithms, such as finding high-order motifs, or simply rerunning algorithms under different experimental conditions, require considerable computational resources. Such graphs and graph algorithms can easily exceed the power of a laptop and/or the patience of the experimenter, but do such graphs exist?

And if we had such large real-world graphs, what would we do with them? What questions would we ask about them? One panelist pragmatically pointed out that the right questions are the ones that have a clear broader impact as defined by the National Science Foundation, the source of much of the funding for graph mining research. Obviously, large graphs allow us to test the scalability of our algorithms, but do we really need trillion-edge graphs to test scalability? Benchmark datasets exist, such as the Graph 500 [29], but the overhead of handling such large graphs becomes an obstacle to the very testing that the benchmarks are designed to support. Also, at some point we must consider the amount of energy necessary to answer the questions we wish to pose. As the area of sustainable computing has been contemplating energy consumption for computation, we as graph miners must also consider the limitations of what is practically computable. Finally, recent efforts in the area of graph stream mining may offer some hope for answering questions once thought intractable on one large graph by streaming the graph in over time.

In the absence of real-world, publicly–available graphs on the order of one trillion in size, one solution is to develop more advanced graph generators that better mimic real-world graph properties. In fact, one of the audience members asked the panelists to comment on the *challenges of generating such synthetic graphs while constraining multiple interdependent graph properties*. Even before we can address this challenge, we need a good model of the distribution of such graphs in the real world, and these models are difficult to obtain [22]. Clearly, no model can represent everything, but which properties are the critical ones to model? It seems that the only way to model real-world networks is to allow them to be built in a realistic way. For example, if you want a model of Wikipedia, then start your on online encyclopedia and monitor its growth. If you want to model email communication, then find a group of people willing to let you monitor their email communication (good luck with that). Currently there are very few robust graph generators, with the exceptions being RMAT [9] and BTER [35]. But RMAT is focused on realistically modeling only the degree distribution. BTER is focused on modeling both degree distribution and triangle distribution, but does a poor job of maintaining a realistic ratio between the two. And neither generator supports the recovery of ground truth, e.g., the true communities for validating community detection algorithms. Furthermore, some panelists pointed out that many algorithms that perform well on these synthetic graphs do not perform well on real-world graphs. The subject of anomalies also came up, and how they can be realistically generated. Manually-constructed anomalies can be inserted into synthetic graphs, but many real anomalies are as yet unimagined. All of this suggests that the proper modeling of real-world graphs, i.e., identifying the salient properties that control the behavior of real-world graphs, and efficiently generating these graphs, remains an important challenge for the field.

4. CONCLUSIONS

The 2015 SDM Workshop on Mining Networks and Graphs provides a valuable snapshot and look ahead for the field. Clearly, the challenge of dealing with large and dynamic graphs is of particular focus, especially choosing proper representations, handing noise, dealing with limited resources, summarization and statistical significance of network mining results. We hope that the workshop proceedings, as well as the summaries of the technical presentations and panel included in this report, will motivate future directions in the field.

5. ACKNOWLEDGMENTS

We would like to thank the workshop program committee for their help selecting a set of quality papers. We would also like to thank the SDM organizers, especially the workshop co-chairs Xiaoli Fern and Xifeng Yan, for their support and guidance in bringing this workshop together. Finally, we would like to thank the authors, panelists and attendees for their contributions and participation.

6. REFERENCES

[1] Abdelhamid, S.H.E.M. et al. 2014. {CINET} 2.0: {A} CyberInfrastructure for Network Science. *10th {IEEE} International Conference on e-Science, eScience 2014, Sao Paulo, Brazil, October 20-24, 2014* (2014), 324–331.

[2] Ahmed, R. and Karypis, G. 2015. Mining Coevolving Induced Relational Motifs in Dynamic Networks. *Proceedings of the 2nd SDM Workshop on Mining Networks and Graphs: A Big Data Analytic Challenge* (2015).

[3] Ahn, K.J., Guha, S. and McGregor, A. 2012. Graph Sketches: Sparsification, Spanners, and Subgraphs. *Proceedings of the 31st Symposium on Principles of Database Systems* (New York, NY, USA, 2012), 5–14.

[4] Bader, J.S., Chaudhuri, A., Rothberg, J.M. and Chant, J. 2004. Gaining confidence in high-throughput protein interaction networks. *Nat Biotech*. 22, 1 (Jan. 2004), 78–85.

[5] Bakshy, E., Rosenn, I., Marlow, C. and Adamic, L. 2012. The Role of Social Networks in Information Diffusion. *Proceedings of the 21st International Conference on World Wide Web* (New York, NY, USA, 2012), 519–528.

[6] Bilgic, M., Mihalkova, L. and Getoor, L. 2010. Active Learning for Networked Data. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (2010).

[7] Burkhardt, P. and Waring, C. 2013. An NSA Big Graph experiment. *Report NSA-RD-2013-056002v1* (2013).

[8] Calderoni, F. 2014. Identifying Mafia Bosses from Meeting Attendance. *Networks and Network Analysis for Defence and Security*. A.J. Masys, ed. Springer International Publishing. 27–48.

[9] Chakrabarti, D., Zhan, Y. and Faloutsos, C. 2004. R-MAT: A Recursive Model for Graph Mining. *SIAM International Conference on Data Mining* (2004).

[10] Cheng, Y. and Wang, N. 2015. Graph clustering by recursive membership identification in neighborhood. *Proceedings of the 2nd SDM Workshop on Mining Networks and Graphs: A Big Data Analytic Challenge* (2015).

[11] Choudhury, S., Holder, L.B., Chin, G., Agarwal, K. and Feo, J. 2015. A Selectivity based approach to Continuous Pattern Detection in Streaming Graphs. *Proceedings of the 18th International Conference on Extending Database Technology (EDBT)* (2015), 157–168.

[12] Dasgupta, A., Kumar, R. and Sarlos, T. 2014. On Estimating the Average Degree. *Proceedings of the 23rd International Conference on World Wide Web* (New York, NY, USA, 2014), 795–806.

[13] Fan, W., Wang, X. and Wu, Y. 2014. Querying Big Graphs Within Bounded Resources. *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data* (New York, NY, USA, 2014), 301–312.

[14] Goldreich, O., Goldwasser, S. and Ron, D. 1998. Property Testing and Its Connection to Learning and Approximation. *J. ACM*. 45, 4 (Jul. 1998), 653–750.

[15] Hasan, S.M.S. et al. 2012. CINET: A Cyberinfrastructure for Network Science. *Proceedings of the 2012 IEEE 8th International Conference on E-Science (e-Science)* (Washington, DC, USA, 2012), 1–8.

[16] Herrera-Semenets, V., Acosta-Mendoze, M. and Gago-Alonso, A. 2015. A Framework for Intrusion Detection based on Frequent Subgraph Mining. *Proceedings of the 2nd SDM Workshop on Mining Networks and Graphs: A Big Data Analytic Challenge* (2015).

[17] Holstein, D., Goltsev, A. V and Mendes, J.F.F. 2013. Impact of noise and damage on collective dynamics of scale-free neuronal networks. *Phys. Rev. E*. 87, 3 (Mar. 2013), 32717.

[18] Huffaker, B., Fomenkov, M. and Claffy, K. 2012. Internet Topology Data Comparison. *CAIDA Technical Report* (2012).

[19] Kashtan, N., Itzkovitz, S., Milo, R. and Alon, U. 2004. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*. 20, 11 (2004), 1746–1758.

[20] Kempe, D., Kleinberg, J. and Tardos, É. 2003. Maximizing the Spread of Influence Through a Social Network. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2003), 137–146.

[21] Khan, M., Klymko, C. and Holder, L.B. eds. 2015. Proceedings of the Second Workshop on Mining Networks and Graphs. *SIAM International Conference on Data Mining* (2015).

[22] Kim, M. and Leskovec, J. 2012. Multiplicative Attribute Graph Model of Real-World Networks. *Internet Math.* 8, 1-2 (2012), 113–160.

[23] Koff, R.S. 1992. Infectious diseases of humans: Dynamics and control. By R.M. Anderson and R.M. May, 757 pp. Oxford: Oxford University Press, 1991. 95.00. *Hepatology*. 15, 1 (1992), 169.

[24] Leskovec, J., Kleinberg, J. and Faloutsos, C. 2007. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*. 1, 1 (Mar. 2007), 1–41.

[25] Li, X., Wu, M., Kwoh, C.-K. and Ng, S.-K. 2010. Computational approaches for detecting protein complexes from protein interaction networks: a survey. *BMC Genomics*. 11, Suppl 1 (2010).

[26] McSherry, F., Isard, M. and Murray, D.G. 2015. Scalability! But at what cost? *15th Workshop on Hot Topics in Operating Systems (HotOS XV)* (Kartause Ittingen, Switzerland, May 2015).

[27] Moreno, S., Kirshner, S., Neville, J. and Vishwanathan, S. 2010. Tied kronecker product graph models to capture variance in network populations. *Allerton'10* (2010), 17–61.

[28] Moreno, S. and Neville, J. 2013. Network Hypothesis Testing Using Mixed Kronecker Product Graph Models. *IEEE 13th International Conference on Data Mining (ICDM)* (Dec. 2013), 1163–1168.

[29] Murphy, R.C., Wheeler, K.B., Barrett, B.W. and Ang, J.A. 2010. Introducing the Graph 500. *Cray User's Group* (May 2010).

[30] Pfeiffer III, J.J., Neville, J. and Bennett, P.N. 2014. Active Exploration in Networks: Using Probabilistic Relationships for Learning and Inference. *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management* (New York, NY, USA, 2014), 639–648.

[31] Prakash, B.A., Chakrabarti, D., Faloutsos, M., Valler, N. and Faloutsos, C. 2011. Threshold Conditions for Arbitrary Cascade Models on Arbitrary Networks. *Proceedings of the 2011 IEEE 11th International Conference on Data Mining* (Washington, DC, USA, 2011), 537–546.

[32] Ron, D. 2010. *Algorithmic and Analysis Techniques in Property Testing*. Now Publishers Inc.

[33] Rubinfield, R. 2006. Sublinear Time Algorithms. *Proceedings of the International Conference of Mathematicians* (2006).

[34] Scaling Apache Giraph to a Trillion Edges: 2013. *https://www.facebook.com/notes/facebook-engineering/scaling-apache-giraph-to-a-trillion-edges/10151617006153920*.

[35] Seshadhri, C., Kolda, T.G. and Pinar, A. 2012. Community structure and scale-free collections of Erdős-Rényi graphs. *Phys. Rev. E*. 85, 5 (May 2012).

[36] Seshadhri, C., Pinar, A. and Kolda, T. 2011. An In-Depth Study of Stochastic Kronecker Graphs. *Proceedings of IEEE International Conference on Data Mining* (2011).

[37] Subramaniyam, N.P. and Hyttinen, J. 2014. Characterization of dynamical systems under noise using recurrence networks: Application to simulated and {EEG} data. *Physics Letters A*. 378, 46 (2014), 3464–3474.

[38] Wang, H., Zheng, D., Burns, R. and Priebe, C. 2015. Active Community Detection in Massive Graphs. *Proceedings of the 2nd SDM Workshop on Mining Networks and Graphs: A Big Data Analytic Challenge* (2015).

[39] Web Data Commons - Hyperlink Graphs: *http://webdatacommons.org/hyperlinkgraph/*.

[40] Zakrzewska, A. and Bader, D.A. 2013. Measuring the Sensitivity of Graph Metrics to Missing Data. *PPAM Workshop on Power and Energy Aspects of Computation* (Sep. 2013).

About the authors:

Lawrence B. Holder is a Professor in the School of Electrical Engineering and Computer Science at Washington State University. He received his Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 1991. Email: holder@wsu.edu.

Maleq Khan is a Research Scientist in the Network Dynamics and Simulation Science Laboratory at Virginia Bioinformatics Institute at Virginia Tech. He received his Ph.D. in Computer Science from Purdue University in 2007. Email: maleq@vbi.vt.edu.

Christine Klymko is a Postdoctoral Researcher in the Center for Applied Scientific Computing at Lawrence Livermore National Laboratory. She received her Ph.D. in Computational Mathematics from Emory University in 2013. Email: klymko1@llnl.gov.

Rajmonda Caceres is a Research Staff Member in the Computing and Analytics Group at the MIT Lincoln Laboratory. She received her Ph.D. in Mathematics and Computer Science from University of Illinois at Chicago in 2012. Email: rcaceres@ll.mit.edu.

Nitesh V. Chawla is a Professor of Computer Science and Engineering at the University of Notre Dame. He received his Ph.D. in Computer Science and Engineering from the University of South Florida in 2002. Email: nchawla@nd.edu.

Tina Eliassi-Rad is an Associate Professor of Computer Science at Rutgers University. She received her Ph.D. in Computer Sciences with a minor in Mathematical Statistics from University of Wisconsin-Madison in 2001. Email: eliassi@cs.rutgers.edu.

David F. Gleich is an Assistant Professor in the Department of Computer Science at Purdue University. He received his Ph.D. in Computational and Mathematical Engineering from Stanford University in 2009. Email: dgleich@purdue.edu.

Ravi Kumar is a Senior Staff Research Scientist at Google, Inc. in Mountain View, CA. He received his Ph.D. in Computer Science from Cornell University in 1998. Email: ravi.k53@gmail.com.

B. Aditya Prakash is an Assistant Professor in the Department of Computer Science at Virginia Tech. He received his Ph.D. in Computer Science from Carnegie Mellon University in 2012. Email: badityap@cs.vt.edu.

Jason Riedy is a Senior Research Scientist in the School of Computational Science and Engineering at the Georgia Institute of Technology. He received his Ph.D. in Computer Science from the University of California Berkeley in 2010. Email: jason.riedy@cc.gatech.edu.

Yinghui Wu is an Assistant Professor in the School of Electrical Engineering and Computer Science at Washington State University. He received his Ph.D. in Computer Science from the University of Edinburgh in 2011. Email: yinghui@eecs.wsu.edu.