Research Statement - Tera-Scale Graph Analysis

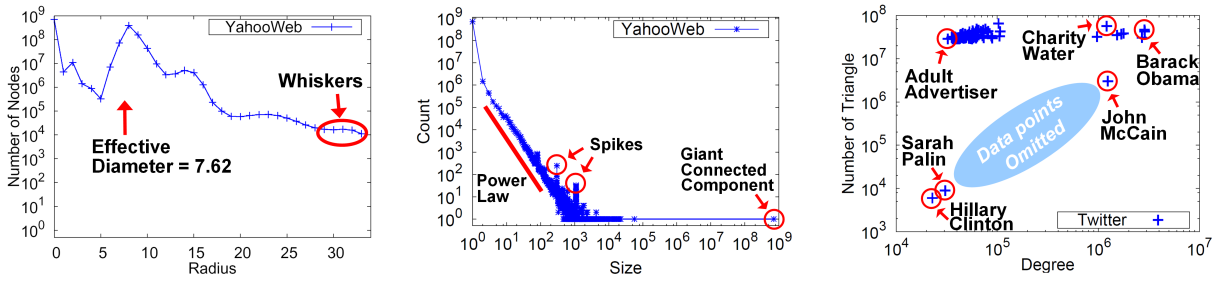
U Kang
Carnegie Mellon University

1 Past and Ongoing Research

My vision is to design and implement *big data analytics system* which finds useful patterns and anomalies in graphs. Graphs are ubiquitous: computer networks, social networks, mobile call networks, protein regulation networks, and the World Wide Web, to name a few. The large volume of available data, the low cost of storage and the stunning success of online social networks and Web2.0 applications all lead to graphs of unprecedented size. Mining large graphs help us discover patterns and anomalies which can be useful for applications ranging from cyber-security (computer networks), fraud-detection (phone companies), and spammer detection (social networks). Typical graph mining algorithms silently assume that the graph fits in the memory of a typical workstation, or at least on a single disk; the above graphs violate these assumptions, spanning multiple Gigabytes, and heading to Tera- and Peta-bytes of data. As a consequence, the vast majority of large graphs has remained untouched.

My research aims to unleash the potential by making extremely scalable graph mining algorithms on distributed platforms. Toward this goal, I have researched on algorithms for scalable graph mining and machine learning analysis. These algorithms are designed to work efficiently on distributed processing platforms including MapReduce and its open source version Hadoop, and analyze patterns and anomalies on very large graphs with more than *billions* of nodes and edges. Specifically, I have worked on the following algorithms.

Structure Analysis. How to find patterns and anomalies in large, real world graphs? Extracting structural features (e.g. radius, diameter, closeness [5], connected components [3], and PageRank scores) is crucial for the discovery of patterns and anomalies in graphs. Previous researches focused on small graphs which fit in a memory or discs of a single machine, but they do not scale to very large graphs with billions of nodes and edges. My goal is to scale up the structure analysis algorithms for very large graphs, and to make the algorithms as general as possible so that we do not reinvent the wheel for similar problems. My approach is to develop an *approximation algorithm* to extract features much efficiently with a high accuracy. I developed Hadi algorithm [8, 9], an approximation algorithm for computing radius and diameter, which provides more than 97% of accuracy but is much efficient than its exact counterpart. Furthermore, I developed an important primitive called `GIM-V` (`G`eneralized `I`terative `M`atrix-`V`ector multiplication) [10, 11] to *unify* many seemingly different feature extraction algorithms. The advantage of the generalization is that an optimization helps boost the performance of many seemingly different algorithms. For example, my block encoding and clustering idea [10, 11] improved the performance by more than *5*× compared to the naive algorithm. Figure 1 (a,b) show the `GIM-V` in action to discover patterns and anomalies in real graphs. Figure 1 (a) displays that the diameter of a very large web graph with billions of nodes and edges is very small, while there exist few anomalous 'whiskers' with large radii. This is



(a) Radius plot of YahooWeb

(b) Connected components of YahooWeb

(c) Triangle counts in Twitter

Figure 1: Patterns and anomalies in large graphs. (a) Radius plot of YahooWeb graph, a Web snapshot at year 2002 containing 1.6 billion pages and 6.6 billion edges. Notice the effective diameter is surprisingly small, and the few whiskers which have large radii. (b) Connected components size distribution of the YahooWeb. Notice the two anomalous spikes which deviate significantly from the constant-slope power law line. (c) The degree vs. participating triangles of some 'celebrities' in Twitter who follows whom snapshot at year 2009. Also shown are accounts of adult sites which have smaller degree, but belong to an abnormally large number of triangles. The reason of the large number of triangles is that adult accounts are often from the same provider, and they follow each other to form a clique, to possibly boost their rankings or popularity.

the first result of the small world phenomenon on the Web. Figure 1 (b) shows the detection of two anomalous spikes which deviate significantly from the constant-slope power law line.

Belief Propagation. Inference in graphs is an important problem, which often corresponds, intuitively, to "guilt by association" scenarios. For example, if a person is a drug-abuser, probably its friends are so, too. The typical way to handle this is belief propagation (BP). My goal is to develop a scalable BP algorithm. The challenge is that naively translating the single machine BP algorithm to distributed streaming algorithm requires a huge, dense matrix. My approach is to develop a *matrix decomposition* method to factor the dense matrix into sparse matrices, and use it to efficiently compute BP on distributed platforms [1, 13]. Thanks to the decomposition, my BP algorithm on HADOOP enjoys *linear scalability* on the number of edges of the input graph.

Eigensolver. How can we find near-cliques, the count of triangles [14], 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. In general, spectral analysis is a fundamental tool not only for graph mining, but also for other areas of scientific computing. My goal is a scalable eigensolver which can handle matrix with more than billions of rows and columns, which existing eigensolvers can not manage. A challenge is to efficiently compute a skewed matrix-matrix multiplication. I believe considering the data distribution is crucial for designing a highly scalable algorithm. My approach is to *exploit the skewness of data distribution* to minimize the amount of data transferred. My skew-aware eigensolver is more than *76*× faster than naive algorithms [4]. Eigensolver can be used to spot anomalous nodes in graphs. For example, I analyzed the Twitter social network using my eigensolver based triangle counting algorithm, and found anomalous adult advertisers with many triangles compared to other nodes with similar degrees, as shown in Figure 1 (c).

Graph Indexing and Compression. Efficiently storing, indexing, and compressing graphs are important to reduce disk storage as well as to answer graph mining queries quickly. My goal is

a node ordering method which groups nonzero elements together, as well as an indexing method to minimize query response time. My approach is to exploit the *power-law* characteristic of real graphs, as well as dynamically select a small subset of compressed edges which might contain the relevant edges to queries. My graph edge layout method [2] outperforms all existing competitors in terms of compression ratio. My graph indexing method [7] provides *50*× smaller storage and *14*× faster running time than without it.

Impact. My research on large scale graph mining has attracted significant interests from academia as well as industries.

- According to Google Scholar, my papers since 2009 have been cited 287 times, with a paper [10] cited 107 times.
- I have packaged my large scale graph mining algorithms into the open-source PEGASUS package (http://www.cs.cmu.edu/∼pegasus), and released it publicly. PEGASUS won the silver award from the Open Source Software World Challenge at 2010, among 26 participants worldwide.
- PEGASUS has been downloaded more than 464 times from 83 countries, and led to 2 U.S. patent applications as well as 2 best paper awards.
- PEGASUS is used as one of the core systems for several DARPA projects including the ADAMS (Anomaly Detection at Multiple Scales) project.
- PEGASUS is officially included in HADOOP for Windows Azure by Microsoft.

2 Future Research

My vision is to design and implement a *big data analytics system* which finds useful patterns and anomalies, and thereby transforming large graphs into valuable sources of knowledge. Toward this goal, I have researched on algorithms for scalable graph mining and machine learning analysis. In the future, I intend to extend the algorithms and systems to handle time evolving graphs, support for near-real time analysis, and apply them to solve many real world problems.

Time Evolving Graphs. Many real world data are time evolving graphs which are naturally modeled as tensors, or multi-dimensional arrays. Examples include dynamic social networks changing over time, hyperlinks and anchor texts in WWW, network traffic over time, and author-conference time-evolving graphs. I plan to generalize my eigensolver to an efficient multi dimensional tensor decomposition algorithm.

Near-Real Time Analytics. Many of my previous works on massive data processing platform have focused on offline batch processing. However some applications (e.g. network attack monitoring) require near-real time responses to queries. I plan to investigate adding real time query response capability to my big data analytics system by quickly combining precomputed statistics and new data.

Big Picture. Finally, I believe my big data analytics system have broad applications beyond the graphs in the computer science contexts. The algorithms that I worked or plan to work on, including belief propagation [1], clustering [12], eigensolver [4], fast kernel [6], time series, and tensor, are fundamental tools useful in various domains of scientific computing. I plan to collaborate with domain experts in astronomy, biology, accounting, health care, environment monitoring, and cyber security to find useful patterns and anomalies. The vision is to focus on real-life applications with massive raw data, and to help extract valuable knowledge.

References

[1] **U. Kang**, Duen Horng Chau, and Christos Faloutsos. Mining large graphs: Algorithms, inference, and discoveries. In *IEEE International Conference on Data Engineering (ICDE)*, pages 243–254, 2011.

[2] **U. Kang** and Christos Faloutsos. Beyond 'caveman communities': Hubs and spokes for graph compression and mining. In *IEEE International Conference on Data Mining (ICDM)*, 2011.

[3] **U. Kang**, Mary McGlohon, Leman Akoglu, and Christos Faloutsos. Patterns on the connected components of terabyte-scale graphs. In *IEEE International Conference on Data Mining (ICDM)*, pages 875–880, 2010.

[4] **U. Kang**, Brendan Meeder, and Christos Faloutsos. Spectral analysis for billion-scale graphs: Discoveries and implementation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 13–25, 2011.

[5] **U. Kang**, Spiros Papadimitriou, Jimeng Sun, and Hanghang Tong. Centralities in large networks: Algorithms and observations. In *SIAM International Conference on Data Mining (SDM)*, pages 119–130, 2011.

[6] **U. Kang**, Hanghang Tong, and Jimeng Sun. Fast random walk graph kernel. In *SIAM International Conference on Data Mining (SDM)*, 2012.

[7] **U. Kang**, Hanghang Tong, Jimeng Sun, Ching-Yung Lin, and Christos Faloutsos. Gbase: a scalable and general graph management system. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1091–1099, 2011.

[8] **U. Kang**, Charalampos E. Tsourakakis, Ana Paula Appel, Christos Faloutsos, and Jure Leskovec. Radius plots for mining tera-byte scale graphs: Algorithms, patterns, and observations. In *SIAM International Conference on Data Mining (SDM)*, pages 548–558, 2010.

[9] **U. Kang**, Charalampos E. Tsourakakis, Ana Paula Appel, Christos Faloutsos, and Jure Leskovec. Hadi: Mining radii of large graphs. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5:8:1–8:24, February 2011.

[10] **U. Kang**, Charalampos E. Tsourakakis, and Christos Faloutsos. Pegasus: A peta-scale graph mining system. In *IEEE International Conference on Data Mining (ICDM)*, pages 229–238, 2009.

[11] **U. Kang**, Charalampos E. Tsourakakis, and Christos Faloutsos. Pegasus: mining peta-scale graphs. *Knowledge and Information Systems (KAIS)*, 27(2):303–325, 2011.

[12] Robson Leonardo Ferreira Cordeiro, Caetano Traina Jr., Agma Juci Machado Traina, Julio López, **U. Kang**, and Christos Faloutsos. Clustering very large multi-dimensional datasets with mapreduce. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 690–698, 2011.

[13] Danai Koutra, Tai-You Ke, **U. Kang**, Duen Horng Chau, Hsing-Kuo Kenneth Pao, and Christos Faloutsos. Unifying guilt-by-association approaches: Theorems and fast algorithms. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 245–260, 2011.

[14] Charalampos E. Tsourakakis, **U. Kang**, Gary L. Miller, and Christos Faloutsos. Doulion: counting triangles in massive graphs with a coin. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 837–846, 2009.