PERSEUS: An Interactive Large-Scale Graph Mining and Visualization Tool

Danai Koutra
Carnegie Mellon University
Pittsburgh, PA
danai@cs.cmu.edu

Di Jin
Carnegie Mellon University
Pittsburgh, PA
dijin@andrew.cmu.edu

Yuanchi Ning
Uber Technologies Inc.
San Francisco, CA
ningyuanchi320@gmail.com

Christos Faloutsos
Carnegie Mellon University
Pittsburgh, PA
christos@cs.cmu.edu

ABSTRACT

Given a large graph with several millions or billions of nodes and edges, such as a social network, how can we *explore* it efficiently and find out *what* is in the data? In this demo we present PERSEUS, a large-scale system that enables the comprehensive analysis of large graphs by supporting the coupled summarization of graph properties and structures, guiding attention to outliers, and allowing the user to interactively explore normal and anomalous node behaviors.

Specifically, PERSEUS provides for the following operations: 1) It automatically extracts graph invariants (*e.g.,* degree, PageRank, real eigenvectors) by performing scalable, offline batch processing on HADOOP; 2) It interactively visualizes univariate and bivariate distributions for those invariants; 3) It summarizes the properties of the nodes that the user selects; 4) It efficiently visualizes the induced subgraph of a selected node and its neighbors, by incrementally revealing its neighbors.

In our demonstration, we invite the audience to interact with PERSEUS to explore a variety of multi-million-edge social networks including a Wikipedia vote network, a friendship/foeship network in Slashdot, and a trust network based on the consumer review website Epinions.com.

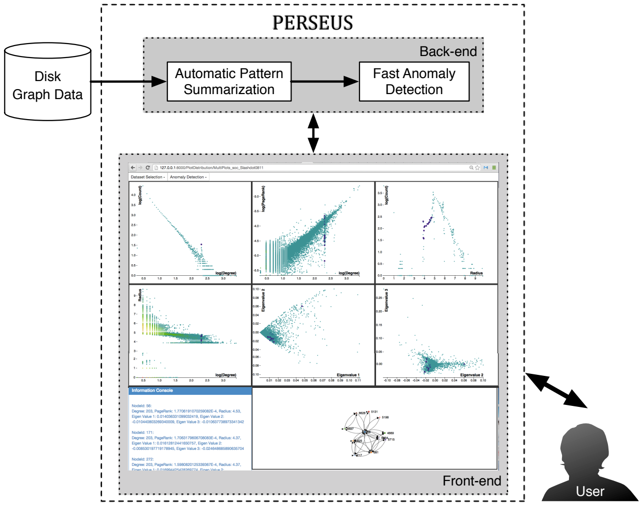


Figure 1: Perseus: system overview. The input graph is automatically processed to summarize graph statistics which are used for offline anomaly detection and visualization. The front-end visualization module, combines the graph properties and anomalies in six linked plots and a dynamic egonet.

1. INTRODUCTION

How can we explore a large graph efficiently and find out what the data can tell us beyond formal modeling and hypothesis testing? In graph mining, although there is often a clear motivation to look at the data and their underlying connections, it is not always clear exactly what one should be looking for. Most traditional methods assume that the user has coding knowledge and/or knows what she is looking for. Thus, they usually focus on one of the following tasks: analysis and modeling of a single graph property; dedicated engines that support querying graphs; speeding up existing graph algorithms; design of anomaly detection algorithms; (interactive) layout-based visualization of a graph. However, often the user does not know how to code, nor what she should be looking for in the data. Instead, she needs to interactively explore a graph and its properties in order to find out *what* is in the data and be able to specify complex questions to ask.

At a high level, PERSEUS is an interactive, large-scale graph mining system that addresses users without programming experience who want to perform guided, preliminary exploration in order to gain insights into their graph data. Our system consists of three main components:

- **Fully-automatic Pattern Summarization**: To summarize the patterns in the input graph we leverage some of the PEGASUS [2] algorithms, which execute in a distributed off-line manner. We fully automate the process of extraction and summarization of graph

Proceedings of the VLDB Endowment, Vol. 8, No. 12
Copyright 2015 VLDB Endowment 2150-8097/15/08.

properties (*e.g.,* as PageRank, radius, degree), which are used both for anomaly detection, and the visualization of distribution plots. This process also generates information about the nodes and the data dependencies, which are used for linking the displayed plots.

- **Fast Anomaly Detection**: In addition to generating distributions of graph properties for visualization purposes, PERSEUS uses the processed data to detect outliers. For this purpose, we leverage G-FADD [5], a fast, density-based anomaly detection method that finds local and global outliers in two or more dimensions. The anomalous candidates are valuable for analysis and attention routing, as they reveal interesting relationship patterns, such as suspicious users (*e.g.,* fraud accounts in Figure 2).
- **Interactive Visualization**: PERSEUS combines the data provided by the first two components into a comprehensive and interactive visualization channel for analysts. It displays univariate and bivariate distributions of the extracted patterns, which may reveal compliance to or deviation from common laws, such as power-law, and guides attention to outliers. PERSEUS also links a selected point to corresponding points in other plots, allowing the user to interactively explore patterns across different distributions. Simultaneously, it provides a summarization of the selected node's properties, its egonet (the induced subgraph of a node and its neighbors), and the properties of some 'similar' nodes, which can be used for further exploration. Thus, by visualizing different feature aspects, PERSEUS provides the users with a global understanding of the normal and anomalous patterns in the data.

Our audience will be invited to interact with PERSEUS on several social and other networks, including a friendship/foeship network based on the technology-related news website named Slashdot[1], which consists of 77, 360 users and about $1M$ edges.

2. SYSTEM OVERVIEW

The following subsections describe in detail how PERSEUS integrates its three main components to support the exploratory analysis of real-world graphs that couples multiple feature aspects, and contributes to a global understanding of the existing patterns.

2.1 Fully-automatic Pattern Summarization

To compute graph statistics efficiently, we leverage the algorithms provided by PEGASUS [2], a Peta-scale graph mining system built on top of HADOOP—an open source implementation of the MapReduce framework which was originally designed for web-scale data processing by Google. PEGASUS proposed an eigensolver for billion-scale, sparse matrices, as well as an optimized primitive for Generalized Iterated Matrix-Vector multiplication, which is a generalization of the plain matrix-vector multiplication and up to 9 times faster than that. In an offline manner, it efficiently computes important graph properties such as degree, PageRank, radius, connected components, and eigenvectors.

PERSEUS fully automates graph processing: It obtains a single graph as input, performs the appropriate transformations depending on its type (*e.g.,* (un)directed) and, for summarization purposes, computes six commonly-used graph properties. These properties include the degree, PageRank, radius, and the first, second and third eigenvectors of the adjacency matrix of the input graph. All of them follow known distributions in real-world graphs. For example, the node degrees follow a power-law-like distribution [1]; there is multi-modal/bi-modal pattern of radius plots [3]; and the eigenvectors of graphs exhibit the "EigenSpokes" pattern [6]. Deviations from those patterns reveal anomalous nodes, such as spammers in social networks.

2.2 Fast Anomaly Detection

For anomaly detection, we use G-FADD [5], a near-linear, density-based outlier detection algorithm, which operates on multi-dimensional clouds of points. G-FADD builds upon the widely-used Local Outlier Factor algorithm (LOF), which detects outlier points by measuring their local deviations (LOF-scores) with respect to their neighboring points. If the LOF-score of a data point is large, then this point is considered anomalous. However, LOF has quadratic runtime due to the large number of duplicate points in distribution plots. G-FADD overcomes this problem in multi-dimensional plots by (a) treating duplicate points as one super-node, and (b) applying a k-dimensional grid on the cloud of points and considering only the cells with sufficient number of supernodes. These optimizations allow G-FADD to run on datasets with tens of millions of points, while LOF runs out of memory even for 20, 000 data points.

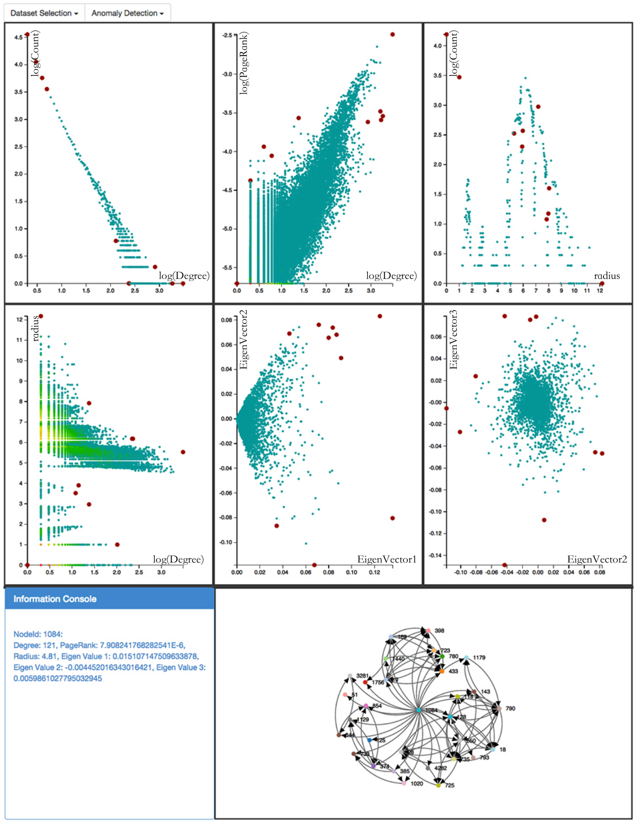
PERSEUS leverages G-FADD to find anomalies in two-dimensional plots that correspond to univariate and bivariate distributions of the pre-computed graph properties: (i) degree, (ii) radius, (iii) degree vs. PageRank, (iv) degree vs. radius, (v) 1^{st} vs. 2^{nd} eigenvector, (vi) 2^{nd} vs. 3^{rd} eigenvector. Traditional methods only focus on one or two distribution plots, but PERSEUS provides a comprehensive view of six plots, and links anomalous points across all of them, allowing an 'ensemble' discovery of patterns. The user can run anomaly detection for different grid sizes (*e.g.,* 0, 8, 16, 32, where 0 finds global outliers and 32 finds local outliers), and PERSEUS annotates the outliers in all the plots. For example, in the first plot (degree distribution) of Figure 2, all the red points correspond to degree-related anomalies for grid size equal to 16. At different granularity, different data points may be flagged as anomalous.

2.3 Interactive Visualization

The front-end visualization component presents six interactive distribution plots and a pane to display egonets. The users can select data points in the plots, find their coordinates, choose to see anomalous points, get the summarized graph properties of one or more selected nodes, and incrementally visualize a node's egonet (bottom right pane in Figure 2). The interactive visualization component consists of two main parts: six linked plots and a dynamic egonet pane.

Linked Plots. The visualization component provides six "linked" univariate and bivariate distribution plots, such as degree distribution, degree vs. PageRank, and eigenvector plots. The user can scroll over the plots, retrieve the coordinates of a specific point, click on it and retrieve the

[1] https://snap.stanford.edu/data/soc-Slashdot0811.html



NodeId: 1084:
Degree: 121, PageRank: 7.908241768282541E-6,
Radius: 4.81, Eigen Value 1: 0.015107147509633878,
Eigen Value 2: -0.004452016343016421, Eigen Value 3:
0.0059861027795032945

Figure 2: Perseus front-end system overview. From top, left to right: i) degree distribution; ii) degree vs. PageRank distribution; iii) radius distribution; iv) degree vs. radius distribution; v) 1^{st} vs. 2^{nd} eigenvector; vi) 2^{nd} vs. 3^{rd} eigenvector. Bottom left: an information console presents a summary of the graph properties for selected nodes. Bottom right: dynamic egonet. The annotated red points correspond to automatically detected outliers.

corresponding points in the other five plots (cyan points in Figure 3(a)). For example, if the user picks a point in the degree distribution plot (*e.g.,* nodes of degree 21), then Perseus randomly selects at the most ten such nodes and highlights their corresponding points in the other distribution plots. The user can also perform anomaly detection at different granularities and the system highlights the anomalous points per plot (red points in Figure 2). As we show in Section 3, the linked plots help the user get the whole picture by connecting the dots across different plots and features, and better understand the data at hand.

Dynamic Egonet. The information console at the bottom left of Figure 2 summarizes the graph properties of one or more selected nodes. The user can select a node from the information console and Perseus generates an incremental visualization of its egonet on the bottom right pane. Many systems attempt to visualize the whole egonet of a node, resulting in a clutter of nodes and edges which is very slow to generate. Perseus avoids this problem by incrementally generating the egonet of a node: when a node is clicked, the next ten highest-PageRank neighbors and their induced subgraph are displayed. The intuitive interaction with the dynamic egonet contributes to further exploration and the understanding of the patterns in the data [4], and complements the distribution-based patterns highlighted by the linked plots.

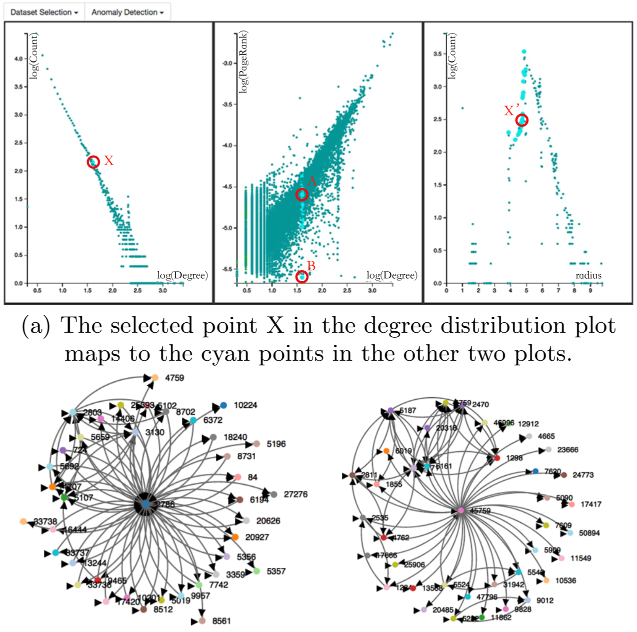
3. ANALYSIS EXAMPLES

Here we showcase the two ways that Perseus can be used, for anomaly detection: (i) user-guided mode, on the Slashdot network[1], and (ii) system-guided mode, on the Wikipedia vote network[2]. The six linked plots and the displayed dynamic egonet help discover several interesting patterns in the above, real-worlds networks.

3.1 User-Guided Anomaly Detection

Here, the user selects a point in one of the distribution plots, and then examines the connections to the other plots and the egonet visualization. Suppose that the user selects point X in the degree distribution (see Figure 3(a), left). Perseus automatically highlights several points in the degree vs. PageRank plot, as well as in the radius plot (Figure 3(a), middle and right). These points correspond to nodes with the user-specified degree.

This type of multi-faceted analysis can reveal anomalies, such as node B in Figure 3(a), which exhibits strange behavior, and specifically in the plot of degree-vs.-PageRank. Without the bivariate distribution plot, node A (id: 2786) and node B (id: 45759) appear to be normal, as their corresponding point (point X) in the degree distribution seems "normal": it fits the power law. Also they are very close to each other in the radius distribution (point X'), which does not raise any suspicions. However, they differ not only in the degree vs. PageRank plot, but also their egonet structure: In our 'user-guided' scenario, the user could click on node 'B' and see its egonet (Figure 3(c)), which looks like a star: 'B' follows many people, who do not reciprocate. In contrast, when the user clicks on node 'A', Perseus responds with Figure 3(b), which exhibits a highly reciprocal behavior, which seems to be the usual/norm, thus making node 'B' a suspicious node.



(a) The selected point X in the degree distribution plot maps to the cyan points in the other two plots.

(b) Egonet of node A.

(c) Egonet of node B.

Figure 3: User-guided anomaly detection.

[2] https://snap.stanford.edu/data/wiki-Vote.html

3.2 System-Guided Anomaly Detection

Here, the user lets Perseus detect outliers, which are annotated as red points in each plot. The user then clicks on a flagged point and examines its corresponding points on the other plots, and/or egonet. For example, while analyzing the wiki-Vote dataset[2], the user may click on node 766 in the second plot of Figure 5, which is flagged as anomalous in the 1^{st} vs. 2^{nd} eigenvector plot too. The plot shows that this node has a large degree and low PageRank, which means that node 766 mainly follows high-PageRank neighbors without being followed by them. Observing the dynamic egonet of the node reveals a 'reciprocity' pattern, which appears in the communities of "famous" nodes: a node follows multiple tightly-knit communities, but few nodes follow him back.

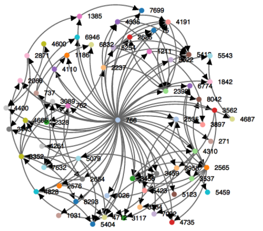


Figure 4: Egonet of anomalous node 766.

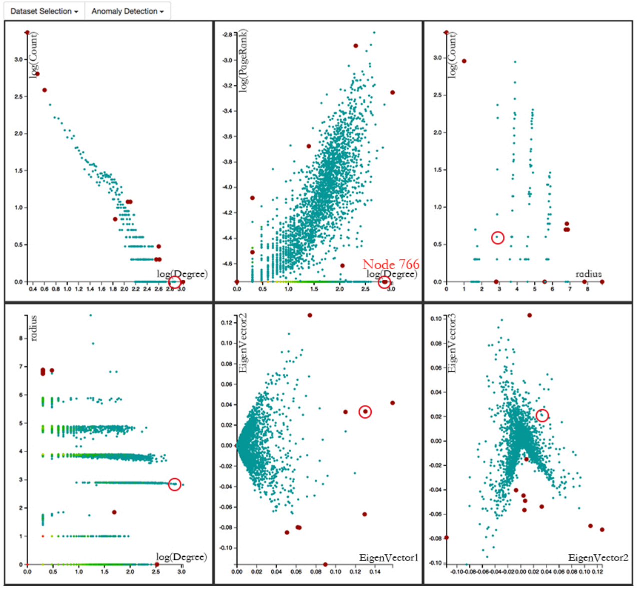


Figure 5: System-guided anomaly detection: The system flags node 766 as anomalous. Its corresponding points are highlighted in red circles.

4. DEMONSTRATION PLAN

In preparation for the demo, the automatic pattern summarization component pre-computes the graph statistics that are used for the univariate and bivariate distributions, and imports them into a Django database. The fast anomaly detection component uses the pre-computed graph statistics to detect anomalous nodes at various granularity levels, and prepares them for visualization. Figure 2 shows a snapshot of the interface with which the audience is invited to interact. The interface includes three main components:

Toolbar. Users can select from a set of datasets that includes the Slashdot graph[1], the Wikipedia vote network[2], and other social networks. Users can also choose the granularity level of G-FADD in order to detect local or global outliers in the displayed distributions.

Linked Plots. Users can mouse over data points (either a point or node of interest, or the marked-in-red anomalies) in the displayed plots to inspect their coordinates, or click on one point to obtain a summary of the node's properties and its egonet. The clicked data point as well as its corresponding points in the other plots get highlighted. The summary of graph properties is displayed in the information console. Selecting nodes in the console triggers the visualization of their egonets.

Egonet. After selecting a node, users can see a summarized version of its egonet, where the node is centered and only ten of its neighbors with the highest PageRank values are displayed. Users can either click on any neighbor to incrementally expand its egonet, or click on the same node again to reveal the next ten highest-PageRank neighbors, as well as the connections between them. For example, Figure 4 shows the egonet of node 766, which consists of its 60 neighbors with the highest PageRank values.

We invite our audience to try out Perseus, and make guided discoveries in large real-world graphs.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1217559 CNS-1314632 IIS-1408924 by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

5. REFERENCES

[1] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM Comput. Commun. Rev.*, volume 29, pages 251–262. ACM, 1999.

[2] U. Kang, C. Tsourakakis, and C. Faloutsos. Pegasus: A peta-scale graph mining system - implementation and observations. *ICDM*, pages 229–238, 2009.

[3] 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.

[4] D. Koutra, U. Kang, J. Vreeken, and C. Faloutsos. VOG: summarizing and understanding large graphs. In *SDM*, pages 91–99, 2014.

[5] J. Y. Lee, U. Kang, D. Koutra, and C. Faloutsos. Fast anomaly detection despite the duplicates. In *WWW companion*, pages 195–196, 2013.

[6] B. A. Prakash, A. Sridharan, M. Seshadri, S. Machiraju, and C. Faloutsos. Eigenspokes: Surprising patterns and scalable community chipping in large graphs. In *PAKDD*, pages 435–448, 2010.