PERSEUS3: Visualizing and Interactively Mining Large-Scale Graphs

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

Ticha Sethapakdi
Carnegie Mellon University
Pittsburgh, PA
tsethapa@andrew.cmu.edu

Danai Koutra
University of Michigan
Ann Arbor, MI
dkoutra@umich.edu

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

ABSTRACT

How can we summarize large graphs of different types, e.g., unipartite or bipartite, directed or undirected? How can we find *anomalous* patterns in such graphs efficiently? In this paper we present PERSEUS3, a large-scale graph mining system that supports analysis of three types of graphs: unipartite and undirected; bipartite and undirected; and unipartite and directed. Our system provides coupled summarization of graph properties and the network structure, and allows the user to interactively explore normal and anomalous node behaviors.

PERSEUS3 is developed based on PERSEUS [7] with three significant extensions: (1) Graph statistics are extracted depending on the type of the input network (*e.g.,* total degree, eigenvectors for undirected graphs; in/out degree, SVD vectors for directed graphs); (2) Subgraphs of the selected node are interactively visualized through the adjacency matrices; (3) Heatmaps (instead of simple scatterplots) are adopted in graph summarization to improve the scalability of the system.

Our extensive experiments show that PERSEUS3 handles different tasks of graph mining efficiently. Specifically we run the univariate undirected graph analysis on a Twitter who-follows-whom graph which spans 0.26 million users and 220 million links; we also run the bipartite graph analysis on a user-movie ratings dataset, and the directed graph analysis on a patent citation graph. We report the patterns discovered, including bipartite cores and outliers spotted by PERSEUS3.

Keywords

Graph summarization; Anomaly detection; Graph mining

1. INTRODUCTION

Modeling and applying algorithms to large graphs is one way of exploring patterns in the data, but, in general, it requires that one should have coding and modeling experience, as well as be aware of what she is looking. Various types of graph analysis nowadays are needed everywhere: web-based communities, such as Facebook and Twitter, are putting more and more effort to perform bot and malicious page detection; the ratings in Netflix need validation that they are from real users; in one's daily life, she might just want to make sure she is not "talking to a dog on the internet". In such scenarios, visualization provides a convenient way to explore big graphs for people with little expertise in graph mining , such as marketing managers, domain experts, and more.

In this paper, we propose PERSEUS3, an interactive, large-scale graph mining system that performs graph summarization and preliminary anomaly exploration and targets users with little experience who want to gain insights into their graph data. PERSEUS3 is built atop PERSEUS [7] which provides the following advantages:

- **Rich types of graph summarization.** PERSEUS3 is capable of handling univariate undirected, bipartite and univariate directed graphs.
- **Interactive subgraph visualization.** Any method that uses graph layouts, will face the "death star problem". That is, if the node belongs to a large clique, its egonet would contain so many edges that it looks chaotic - colloquially called the 'death star'. Our solution is to give the adjacency matrix, instead of the spring-model graph - then, cliques are full areas of the matrix, easily understood by the human analyst.
- **Heatmap representation.** Scatter-plots with millions of points are prohibitively slow to plot. Instead, PERSEUS3 uses heatmaps, achieving up to 30x improvement on the speed of interaction, and making the plotting time effectively constant on the size of the graph.

2. BACKGROUND

Our work is inspired by different fields of research, with the two major ones being: (i) large-scale graph visualization, and (ii) outlier detection.

2.1 Large-scale graph visualization

Apolo [3] is a visualization tool that supports incrementally revealing neighbors of some selected nodes in a graph. NET-RAY [5], a visual mining system, is proposed to handle the visualization of billion-scale graphs, adjacency matrix mining and outlier detection. Although this work provides algorithms to obtain plots for various graph mining tasks, the system is not interactive, thus requiring experienced users to make sense of the results. PERSEUS [7] introduced an interactive large-scale graph visualization and mining

WOODSTOCK '97 El Paso, Texas USA

DOI: 10.475/123_4

system, which supports user attention routing to outliers and interaction with distribution plots. However, it suffers from scalability as the amount of points that need displaying in some distributions of graph properties may overflow the resolution of a typical screen. Also, it only handles univariate, undirected graphs by displaying the univariate distribution of the total node degree, and bivariate distributions of Ritz eigenvectors by symmetrizing the input directed graphs.

2.2 Outlier detection

Many works are proposed in large-scale outlier detection, such as LOF [2] and LOCI [9]. In industry, some practical frameworks are also presented: to detect fake accounts, Facebook immune system was proposed to detect single agents controlling many accounts; CopyCatch [1], a Hadoop-based method was proposed to detect groups of users who coordinate to give page likes. There are also anti-phishing and anti-malware mechanisms, rendering real accounts difficult to be compromised. However, these techniques are either specific to Facebook users, or generally do not support data visualization and user interaction.

In general, there is not much work on large-scale interactive visualization of heterogeneous graphs that also supports outlier detection.

3. METHOD

In this section we describe in detail how PERSEUS3 supports analysis of real-world graphs that couples various components contributing to both a global and local understanding of the existing patterns. An overview of the system is illustrated in Figure 1.

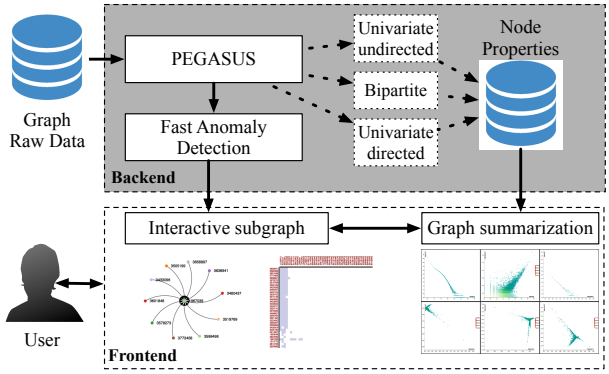


Figure 1: PERSEUS3 system overview. Backend: Graph statistics are extracted depending on the graph type, and stored in the database to handle queries from the frontend. Anomaly detection is also performed based on the graph statistics. Frontend: The user interacts with the graph through dynamic egonet and adjacency matrix of the selected node, along with coupled distributions of graph statistics provided by the graph summarization module.

3.1 Rich Types of Graph Summarization

PERSEUS3 automatically computes various graph properties using PEGASUS [6] in a distributed off-line manner. It then extracts graph statistics depending on the type of the input graph and constructs plots for univariate and/or bivariate relationships.

For univariate, undirected graph analysis, PERSEUS3 converts the input graph to symmetric in order to compute the total degree distribution and Ritz eigenvectors. For unipartite and bipartite *directed* graphs, PERSEUS3 computes the importance of the nodes through the authority and hubness metrics. Such statistics can be calculated through Singular Vector Decomposition (SVD) of the adjacency matrix of the input graph. We use the left singular vectors (U) to measure the hubness and the right singular vectors (V) to measure the authority. In addition, we separate the total degree centrality applied for undirected graphs into in- and out-degree centrality for directed graphs. The complete statistics extracted for all three types of graphs are summarized in Table 1.

Graph type	Statistics
Unipartite + undirected	Total degree, PageRank, 1^{st}, 2^{nd}, 3^{rd} and 4^{th} eigenvector
Bipartite + directed	In degree, 1^{st}, 2^{nd} V vector (V1, V2), out degree, 1^{st}, 2^{nd} U vector (U1, U2)
Unipartite + directed	In degree, V1, V2 vector, out degree, U1, U2 vector

Table 1: Statistics visualized for each type of graph

The extracted statistics are precomputed and combined into six univariate and/or bivariate distributions displayed in the frontend to provide a global understanding of the normal and anomalous patterns in the data. For undirected graphs, these distributions are: (i) total degree distribution, (ii) total degree vs. PageRank, (iii) PageRank distribution, (iv) 1^{st} vs. 2^{nd} eigenvector, (v) 2^{nd}vs. 3^{rd} eigenvector and (vi) 3^{rd} vs. 4^{th} eigenvector. For directed graphs, the six distributions are: (i) in-degree distribution, (ii) in-degree vs. V1 vector, (iii)V1 vs. V2 vector, (iv) out-degree distribution, (v) out-degree vs. U1 vector and (vi) U1 vs. U2 vector. In each type of graph analysis, the points in all the distribution plots are linked through their corresponding nodes in the backend database. For instance, if the user clicks on a point in a specific plot the system highlights the corresponding points in the remaining five plots. Note that for bipartite graphs, as the source nodes do not connected to each other via edges, and neither do the destination nodes, the distributions (i)-(ii)-(iii) and (iv)-(v)-(vi) are linked separately for bipartite graphs.

All the distributions considered follow known distributions in real-world graphs, *e.g.*, the degree distribution and the PageRank distribution follow a power-law-like pattern; the distributions of eigenvectors exhibit the "EigenSpokes" [10] pattern if tightly connected components exist, and intuitively in bipartite graphs, nodes with high value of in/out degree tend to have high values in V/U vectors as such nodes tend to be important destination/source nodes.

3.2 Interactive Subgraph Visualization

In addition to the dynamic egonet, PERSEUS3 includes the interactive adjacency sub-matrix representing the subgraph containing the selected node and its 1-hop neighbors to provide the user with local understanding. For unipartite, undirected or directed graphs, the rows of the adjacency sub-matrix are sorted by the value in the first left singular vector (U1), and the columns are sorted by the value in the first right singular vector (V1). For nodes with more than 100 neighbors, the top 100 of them and their edges are displayed. The richer neighboring information brought by the adjacency matrix could guide the interaction with the egonets so that the user can detect group anomalies such as bipartite cores, which solves the "death star problem" mentioned in the introduction.

However, this approach provides little information for bipartite graphs as the 1-hop neighbors are all either source nodes or destination nodes, rendering only one row or column in the adjacency matrix. To handle this problem, PERSEUS3 identifies similar nodes based on common neighbors to the one that the user selects, and visualizes them along with their edges in the corresponding adjacency sub-matrix. To efficiently find similar nodes, PERSEUS3 employs Local Sensitivity Hashing[11] (LSH) to pre-compute the

similarity of pairs of nodes as it avoids the quadratic computational cost. LSH computes and sorts pairs of nodes based on the number of common neighbors and in bipartite graphs, similar pairs of source/destination nodes are found based on common destination/source nodes. If a user clicks on a source node, up to 100 similar source nodes are displayed as rows sorted by U1 in the adjacency matrix and up to 100 destination nodes they are pointing to are displayed as columns sorted by V1. Similar rules apply when a user clicks on a destination node. Figure 2 shows an example of bipartite graph analysis, where PERSEUS3 displays the adjacency matrix for the node 1940, which was selected in the U1 vs. U2 distribution plot. The high density of the row that corresponds to node 1940 indicates that the corresponding user is active and rates many movies. The adjacency sub-matrix also displays other users based on common rated movies. The adjacency sub-matrix 'links' to all the other plots–e.g., if node 1979 is of interest and the user clicks on it, the corresponding points will be highlighted in the distribution plots.

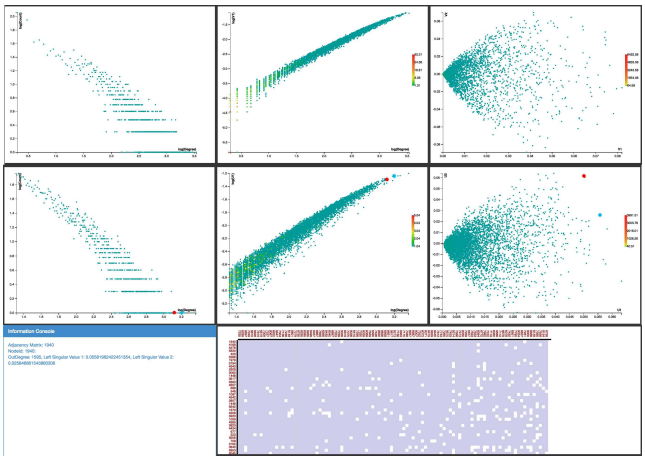


Figure 2: "Normal" graph (MovieLens, 1M) exhibits power laws, and no micro-clusters in PERSEUS3. From top, left to right: 1) in-degree distribution; 2) in-degree vs. V1; 3) V1 vs. V2; 3) out-degree distribution; 4) out-degree vs. U1; 5) U1 vs. U2. Bottom left: information console with a summary of the selected node's properties. Bottom right: the interactive adjacency matrix of node 1940. The cyan points correspond to the selected node (1940). The red dots correspond to node 1979 which was clicked in the adjacency matrix.

3.3 Heatmap Representation

PERSEUS3 widely uses the idea of heatmaps, and points with identical graph statistics are aggregated in the distribution plots of graph properties. This handles nicely exact-duplicate points, but fails to address the slow projection time for most distributions, including eigenvectors (for undirected graphs) and singular vectors (directed), as their values are calculated with accuracy of 10 decimal places. Plotting all the points unnecessarily burdens the frontend, which has to project millions of dots although they are too close to each other to be distinguished by a human eye. PERSEUS3 addresses this problem by dividing the plot into a $k \times k$ grid (the default $k = 1000$) and *then* computing the heatmap: grid cells ('super-points') with many points, become more red. Thanks to our optimization, we can achieve 20x time savings or more: we need less than 1 second, to display a plot of 77K points[1], which would normally take 28 seconds (unacceptable, for human interaction).

4. EXPERIMENTS

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

To demonstrate that the proposed method is generic and can be applied in different contexts, we perform experiments on three heterogeneous datasets and address three research questions: (1) What do extreme points denote in the eigenvector distributions? (2) What do the graph property distributions look like for a normal graph and one with anomalies? (3) What anomalous patterns does PERSEUS3 find in real-world graphs?

4.1 Univariate Undirected Graph Analysis

Twitter: President election This dataset [8] contains 126,628 accounts and 4,191,918 tweets. Each recorded tweet is either related to 2012 presidential election or posted by users who were active on that topic. We use the notation @<username> to denote a user. The graph is constructed by treating accounts (or users) as nodes and the who-retweets-whom relationships between two accounts as links. The frontend of PERSEUS3 is shown in Figure 3.

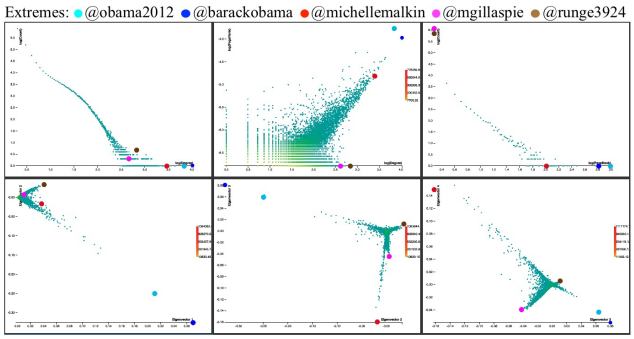


Figure 3: Graph summarization of the Twitter dataset with colored dots correspond to nodes that make sense. Blue and cyan: President Obama (democrat); red: Michelle Malkin (conservative commentator); pink: mgillaspie (tea partier) and brown: runge3924 (suspicious account). PERSEUS3 helps spot (at least) 4 groups / spokes.

From top, left to right we illustrate the (1) total-degree distribution; (2) total-degree vs. PageRank; (3) PageRank distribution; (4) 1st vs. 2nd eigenvector, (5) 2nd vs. 3rd eigenvector; and (6) 3rd vs. 4rd eigenvector distribution. We are interested in the "spike" patterns in the eigenvector distributions, so we explore the details of five selected nodes, marked in blue, cyan, red, pink and brown in Figure 3. By leveraging the backend database, we found some interesting patterns:

1. We observed that the blue node is close to the cyan one in every distribution. It turns out to be two accounts relevant to the same person.

2. In plot (4), there are two spikes with extremes @runge3924 and @barackobama. According to the context lookup, user @runge3924 has 1237 retweets, but were retweeted 0 times. In contrary, the number of retweets and retweeted messages of @barackobama both rank top.

3. Representing different political opinions, @michellemalkin (conservative commentator), @mgillaspie (tea partier) and @barackobama (democrat) along with @runge3924 are located at the extreme points of 4 spikes in plots (5) and (6).

For pattern 1, it means that accounts with similar purposes, such as @barackobama and @obama2012, share almost identical statistics in all of the distribution plots, as their retweet behaviors are basically the same.

In pattern 2, we suspect user @runge3924 to be a bot in Twitter as it only retweets others and never gets retweeted. On the contrary, active accounts, such as @barackobama and @obama2012, form a community whose posts are being retweeted. These contradictory behaviors explain why their corresponding points have opposite locations in the eigenvector plots.

Pattern 3 reflects the communities formed by users with different political opinions. Real users in Twitter tend to interact with people sharing the same interests, and form communities with different topics. Since users of this dataset mainly focus on politics, different political communities are detected. The bots, however, form totally different communities from the majority of users in the graph.

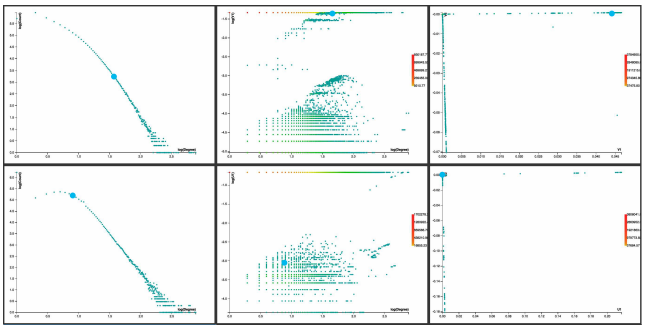
4.2 Bipartite Graph Analysis

MovieLens 1M This dataset[4] contains 1,000,209 anonymous ratings of approximately 3,900 movies given by 6,040 users who joined MovieLens in 2000. The frontend is shown in Figure 2.

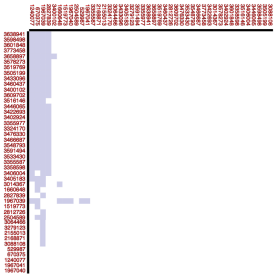
In this bipartite graph, we see that almost all the nodes comply with common laws discussed above. This could indicate that there are no anomalous users in this "stable benchmark dataset", which can be used as a reference for 'normal' graph summarization.

4.3 Univariate Directed Graph Analysis

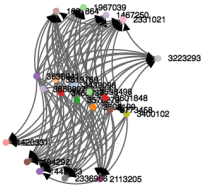
Patent citation This dataset[2] contains 3,774,768 unique patents and 16,518,948 directed citations among them. The graph summarization is shown in Figure 4 (a).



(a) Graph summarization of the patent citation dataset with selected node marked in cyan



(b) Adjacency matrix



(c) Egonet

Figure 4: PERSEUS3 on a directed graph (patent citation)

Clearly there is an "eigenspoke" pattern in the U and V distributions, indicating the existence of bipartite cores. By clicking nodes in the spike, *e.g.,* the one marked in cyan in Fig. 4 (a), PERSEUS3 returns its adjacency matrix illustrated in Fig. 4 (b), which exhibits the clear pattern of bipartite cores. Led by this information, further exploration on the egonet can be conducted. By expanding the neighbors of the selected node in the egonet we confirm that it belongs to a bipartite core, shown in Fig. 4 (c). Compared with (b) and (c), we can find the former plot exhibits the clear pattern of a bipartite core, while the latter looks messy.

[2]https://snap.stanford.edu/data/cit-Patents.html

5. CONCLUSIONS

In this paper, we presented PERSEUS3 which tackles large-scale graph mining in an interactive manner. PERSEUS3 supports the analysis of three different types of graphs, and helps with both global and local understanding of *normal* and *anomalous* patterns in the data through rich types of graph summarization and interactive subgraph visualization. The heatmap representation that PERSEUS3 adopts guarantees the ability to display plots with millions of points, while maintaining the useful patterns. Moreover, we showed how to use our system to visualize large, real-world graphs and reported interesting discoveries and anomalies found, including extreme points and near-bipartite cores and spikes.

Reproducibility: Our code is open-sourced and can be found at https://www.dropbox.com/s/rl7m5yro8bgvx40/PERSEUS_light.zip?dl=0 along with the twitter dataset (used in Figure 3).

6. 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.

7. REFERENCES

[1] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos. Copycatch: stopping group attacks by spotting lockstep behavior in social networks. In *WWW*, pages 119–130, 2013.

[2] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000.

[3] D. H. Chau, A. Kittur, J. I. Hong, and C. Faloutsos. Apolo: interactive large graph sensemaking by combining machine learning and visualization. In *KDD*, pages 739–742, 2011.

[4] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM TiiS*, 5(4):19, 2015.

[5] U. Kang, J.-Y. Lee, D. Koutra, and C. Faloutsos. Net-ray: Visualizing and mining billion-scale graphs. In *Advances in Knowledge Discovery and Data Mining*, pages 348–361. Springer, 2014.

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

[7] D. Koutra, D. Jin, Y. Ning, and C. Faloutsos. Perseus: an interactive large-scale graph mining and visualization tool. *PVLDB*, 8(12):1924–1927, 2015.

[8] Y.-R. Lin, B. Keegan, D. Margolin, and D. Lazer. Rising tides or rising stars?: Dynamics of shared attention on twitter during media events. *PloS one*, 9(5):e94093, 2014.

[9] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *ICDE*, pages 315–326. IEEE, 2003.

[10] 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.

[11] M. Slaney and M. Casey. Locality-sensitive hashing for finding nearest neighbors [lecture notes]. *Signal Processing Magazine, IEEE*, 25(2):128–131, 2008.