AdaptiveNav: Discovering Locally Interesting and Surprising Nodes in Large Graphs

Robert Pienta, Zhiyuan Lin, Minsuk Kahng, Jilles Vreeken
Partha P. Talukdar, James Abello, Ganesh Parameswaran, Duen Horng (Polo) Chau

Abstract—
Visualization is a powerful paradigm for exploratory data analysis. Visualizing large graphs, however, often results in a meaningless hairball. In this paper, we propose a different approach that helps the user *adaptively* explore large million-node graphs from a *local* perspective. For nodes that the user investigates, we propose to only show the neighbors with the most subjectively interesting neighborhoods. We contribute novel ideas to measure this interestingness in terms of how surprising a neighborhood is given the background distribution, as well as how well it fits the nodes the user chose to explore. We are currently designing and developing AdaptiveNav, a fast and scalable method for visually exploring large graphs. By implementing our above ideas, it allows users to look into the forest through its trees.

Index Terms—Graph Visualization, Data Exploration, Adaptivity, Serendipity

1 Introduction

Visualization is perhaps the most natural approach to exploratory data analysis. Under the right visualization, finding patterns, deciding what is interesting, what is not, and what to investigate next are all easy tasks – in a sense the answers 'jump to us' as our brains are highly specialized for analyzing complex visual data. Visualizing large graphs in an intuitive and informative manner has proven to be rather difficult. Even using advanced layout techniques plotting a graph typically leads to a useless "hairball" from which nothing can be deduced [4, 5]. Instead of plotting the whole graph, visualizing only part of the graph seems more promising [7, 2]. Doing this naively leads to the same problem. Because real world graphs are often scale free (follow a power law degree distribution [3]), even a single hop expansion from a selected node can be visually overwhelming.

We take a different approach. We propose to *adaptively* explore large graphs from a *local* perspective. That is, starting from an initially selected node – e.g., explicitly queried by the user, or proposed by algorithms [1, 6] – we propose to only show the most interesting neighbors. We identify these by their *subjective interestingness*, by how surprising their neighborhood distribution is (e.g., do neighbors' degree distributions follow a power law like when considering all nodes?), as well as by how similar this distribution is to those of the nodes the user already visited during the exploration. By only showing those parts of the graph that are most informative to the user, we keep the view clean.

We call our adaptive approach AdaptiveNav — our idea is a significant addition to existing works that aim to recommend individual nodes to users; instead, we steer users towards local regions of the graphs that match best with their current browsing interests, helping them better understand and visualize the graph at the same time. By being adaptive, AdaptiveNav allows every user to explore those facets of the graph that are most interesting to them.

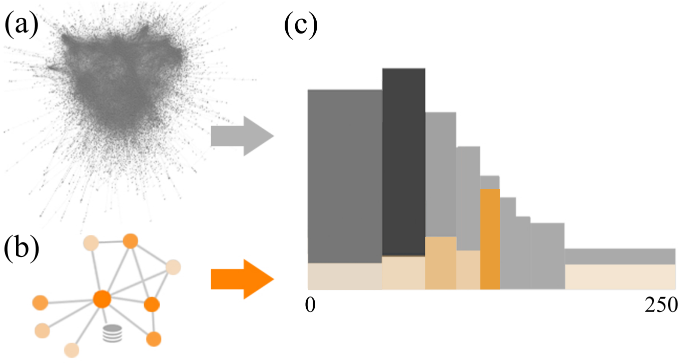


Fig. 1. Local and global distribution histograms are both essential to AdaptiveNav. Local histograms (orange bars) are representations of feature distributions (e.g., neighbors' degree distribution) in a single node's egonet. The global distributions (gray bars) depicts the corresponding feature's distribution across the whole graph. The difference between those two distributions reflects if a node is ”unusual” or surprising compared to the majority in the graph.

2 FACET

2.1 Model

AdaptiveNav guides users during their graph exploration using both surprisingness, which has been shown as an effective way to introduce new insights to users [6], and subjective interest that changes dynamically to suit the user, which significantly differs from previous works (e.g., [7] uses "degree of interest") that only offers static recommendations (i.e., do not actively adapt to users' change of behaviors).

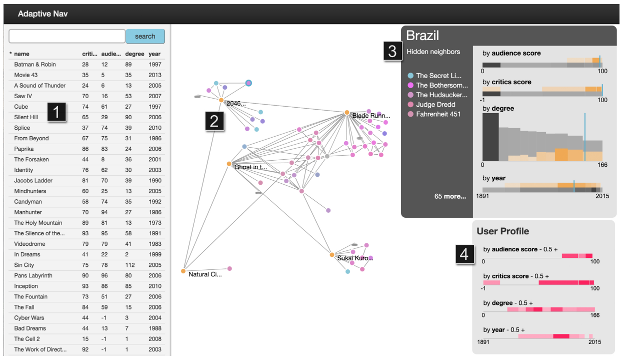
The surprisingness rankings are determined by the divergence of local feature distributions from the background feature distributions (e.g., local degree distribution vs. global degree distribution). Through our experiments we have chosen Jensen-Shannon (JS) divergence, a symmetrical version of Kullback-Leibler divergence to construct our surprisingness metric. JS divergence works well, because the resulting output is in bits so the divergences of several features can be easily combined into a single score. We measure surprise by determining the divergence of feature distributions $L_{i,j}$ over a node's neighborhood V_a (1 hop), from the global distributions of features G (see Equation 1).

$$s_i = \sum_{f_j \in A} \lambda_j D_{JS}(L_{i,j} || G_j), \qquad (1)$$

where L_j and G_j are the local and global distributions of node-feature f_j and λ_j is a feature weight.

- *Robert Pienta, Zhiyuan Lin, Minsuk Kahng, and Duen Horng (Polo) Chau are with Georgia Institute of Technology.*
- *Jilles Vreeken is with Max Planck Institute for Informatics and Saarland University.*
- *Partha P. Talukdar is with Indian Institute of Science.*
- *James Abello is with Rutgers University.*
- *Ganesh Parameswaran is with Yahoo! Inc.*

Manuscript received 31 Mar. 2014; accepted 1 Aug. 2014; date of publication xx xxx 2014; date of current version xx xxx 2014. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.



* name	criti...	audie...	degree	year
Batman & Robin	28	12	89	1997
Movie 43	35	5	35	2013
A Sound of Thunder	24	6	13	2005
Saw IV	70	16	53	2007
Cube	74	61	27	1997
Silent Hill	65	29	90	2006
Splice	37	74	39	2010
From Beyond	67	75	31	1986
Paprika	86	83	24	2006
The Forsaken	44	8	36	2001
Identity	76	62	30	2003
Jacobs Ladder	81	70	39	1990
Mindhunters	60	25	13	2005
Candyman	58	74	35	1992
Manhunter	70	94	27	1986
The Holy Mountain	89	81	13	1973
The Silence of the...	93	95	58	1991
Videodrome	79	79	41	1983
In Dreams	41	22	2	1999
Sin City	75	78	112	2005
Pans Labyrinth	90	96	80	2006
Inception	93	86	85	2010
The Fountain	73	51	27	2006
The Fall	84	59	15	2006
Cyber Wars	44	-1	3	2004
Bad Dreams	44	13	7	1988
The Cell 2	15	-1	1	2008
The Work of Direct...	92	-1	1	2003

Fig. 2. The AdaptiveNav user interface displaying an explored portion of the RottenTomatoes similar-movie graph. The user has traversed several films (shown as orange nodes with titles) and AdaptiveNav has displayed a subset of the relevant neighbors (the nodes with colors ranging from blue to red) and their connectivity. Figures are viewed best in color.

The subjective interest rankings arise from the divergence between the local feature distributions and the user's current profile, which represents their exploration, in node-features, up to that point. We track the user's behavior and record a user profile as they explore their data. Each clicked node offers valuable details into the types of nodes in which the user is interested. This forms distributions U_j for each feature f_j. To rank the user's interest in the undisplayed neighbors of node v_a we follow a similar approach as Equation 1.

In summary, whenever a user selects a node to explore, we rank its neighbors based on surprise and subjective interestingness. For each of the neighbors, we compute surprise and interest scores for each feature and aggregate them based on feature weights λ_j. We blend those scores, and finally, nodes with k highest scores will be returned.

2.2 Components

AdaptiveNav's user interface as shown in Figure 2 has four key elements: The first main area is the **Table View** (1), a conventional table showing the currently displayed nodes and some of their features. This provides sortable node-level information. The central area is the **Graph View** (2). It is an interactive *force-directed graph layout* that demonstrates the structure and relationships among nodes as the user explores. Coloring the nodes is used to encode the surprise and interest based on the user's current exploration. The **Neighborhood Summary** (3) allows a user to investigate the feature distributions of its currently undisplayed neighbors. It presents the user with feature *heat maps* that summarize the distributions of hidden nodes, which can be clicked and turn into *histograms*. This lets a user quickly select new nodes based on their feature values and get a quick summary of this node's neighborhood. As a user explores, we construct and display a summary profile of the important features they have covered in the **User Profile** view (4). The user profile view suggests high-level browsing behavior to the user; allows for better understanding of where the user-interest ranking comes from; and allows them to adjust if they want to ignore certain features in the interest ranking.

3 Scenario

To illustrate how AdaptiveNav works in practice, consider our user Susan who is looking for interesting movies to watch (see Figure 3), by exploring a Rotten Tomatoes movie similarity graph with 17k movies. In this graph, an edge connects two movie nodes if users of RottenTomatoes voted them as similar films. Susan has watched *Miss Congeniality* , a crime-comedy that stars Sandra Bullock as an FBI agent who thwarts terrorist efforts by going undercover, turning her rude unflattering self into a glamorous beauty queen. AdaptiveNav also suggests a few movies that are interesting and surprising to her.

Matching Susan's interest, AdaptiveNav suggests the *Big Mommas House* series, which also have undercover plots and are interestingly like *Miss Congeniality*. They both share low critics scores, but high audience scores. To Susan's surprise, AdaptiveNav suggests

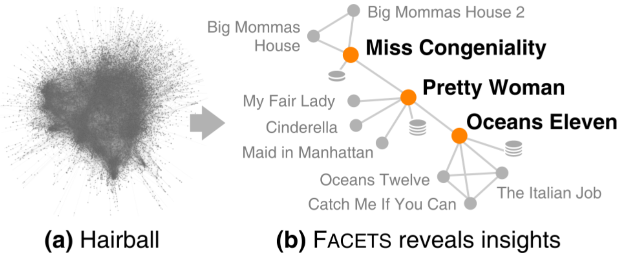


Fig. 3. (a) The Rotten Tomatoes movie graph shown using conventional spring layout (an edge connects two movie nodes if some users voted them as similar). Even for this relatively small graph of 17k nodes and 72k edges, a global visualization does not provide much insight. (b) A better way, using our AdaptiveNav approach, focuses on movies that are the most subjectively interesting, surprising, or both. For example, AdaptiveNav suggests *Pretty Woman* (romantic-comedy) as a interesting, suprising related movie of *Miss Congeniality* (crime-comedy).

Pretty Woman, which is quite different (thus surprising) — a romantic-comedy that has both scores from the critics and the audience. But, there is still more subtle similarity (thus still drawing Susan's interest); both films share a Cinderalla-like storyline, which explains why the two movies are connected in the graph: Sandra Bullock goes from a rude agent to a beauty queen; in *Pretty Woman*, Julia Roberts goes from a prostitute to a fair lady. In fact, *Pretty Woman* is a classic, exemplar romantic-comedy; many movies follow similar story lines (e.g., *Maid in Manhattan*). Thus, *Pretty Woman* has very high degree in the graph, unlike *Miss Congeniality* which is a niche genre; this also contributes to *Pretty Woman*'s surprisingness.

Through *Pretty Woman*, AdaptiveNav again pleasantly surprises Susan with *Oceans Eleven*, which also stars Julia Roberts, and is in a rather different light hearted crime or heist genre, introducing Susan to other very similar movies like *Oceans Twelve* and *The Italian Job*. Figure 3(b) summarizes Susan's exploration. If Susan were to use a conventional visualization tool to perform the same kind of movie exploration, she would likely be completely overwhelmed with an incomprehensible hairball visualization (as in Figure 3(a)).

4 Conclusion

In this work, we describe AdaptiveNav, an integrated approach that combines visualization and computational techniques to help the user performs adaptive exploration of large graphs. AdaptiveNav overcomes many issues commonly encountered when visualizing large graphs. AdaptiveNav shows the user only the most subjectively interesting material as they explore. We do this by ranking the neighbors of each node by surprisingness and subjective interest based on what the user has explored so far. The user interest ranking influences the direction of a user's exploration by showing the best matches to their current taste in nodes. Despite the old adage, you can see the graph through its nodes.

References

[1] L. Akoglu, M. McGlohon, and C. Faloutsos. Oddball: Spotting anomalies in weighted graphs. In *PAKDD*, pages 410–421. Springer, 2010.

[2] 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. ACM, 2011.

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

[4] D. A. Keim. Visual exploration of large data sets. *Commun. ACM*, 44(8):38–44, 2001.

[5] D. A. Keim. Information visualization and visual data mining. *IEEE TVCG*, 8(1):1–8, 2002.

[6] K. Onuma, H. Tong, and C. Faloutsos. Tangent: a novel,'surprise me', recommendation algorithm. In *KDD*, pages 657–666. ACM, 2009.

[7] F. Van Ham and A. Perer. Search, show context, expand on demand : Supporting large graph exploration with degree-of-interest. *IEEE TVCG*, 15(6):953–960, 2009.