Adaptive Visualization of Research Communities

Martijn de Jongh, Patrick M. Dudas, and Peter Brusilovsky

University of Pittsburgh
{mad159,pmd18,peterb}@pitt.edu

Abstract. *Adaptive visualization approaches attempt to tune the content and the topology of information visualization to various user characteristics. While adapting visualization to user cognitive traits, goals, or knowledge has been relatively well explored, some other user characteristics have received no attention. This paper presents a methodology to adapt a traditional cluster-based visualization of communities to user individual model of community organization. This class of user-adapted visualization is not only achievable, but expected due to real world situation where users cannot be segmented into heterogeneous communities since many users have affinity to more than one group. An interactive clustering and visualization approach presented in the paper allows the user communicate their personal mental models of overlapping communities to the clustering algorithm itself and obtain a community visualization image that more realistically fits their prospects.*

Keywords: Community detection, user model, social network, adaptive visualization

1 Introduction

The increased popularity of social networking research attracted attention to the problem of community discovery and visualization. Since the work by Girvan & Newman [1] many different approaches to discover communities (i.e., clusters of similar users) in social networks and other social systems were suggested and explored - see Fortunato [2] for a comprehensive overview. In addition, a number of packages such as Gephi [3], Pajek [4], and Ucinet [5] were developed to visualize the results of community discovery to the end users.

Despite a relatively large volume of work on the topic, little attention was paid to take into account user mental models and domain knowledge when presenting visual structure of the community. Existing visualization programs tend to represent a simplified community organization formed by a number of distinct, non-overlapping communities that are displayed universally to all users of the system.

The novelty of our approach is the understanding that different users can form different models of community organization. They can recognize different subcommunities within the same large community due to their unique personal knowledge and domain expertise. For example, a researcher working on the application of machine learning to user modeling could be considered as a user modeling researcher by one group of users and as a machine learning researcher by another. Existing visualizations do not recognize these individual preferences and as a result produce community structure that might be acceptable only by a subset of the target users.

This paper presents our adaptive platform to create an interactive community visualization that can take into account user preferences on community organization and provide dynamic adaption to the user model of community structure. To collect user preferences, the system allows the users to identify cliques of researchers that, from their prospect, should belong to the same community. These user-defined cliques are considered by an interactive clustering approach developed by us along with the original data, describing similarity between researchers, to produce a user-adapted cluster visualization.

The presentation of our approach is organized as follows. We start with the interface part of the approach explaining how our system allows the users to specify their preferences. Then we provide the methodology of our approach explaining the interactive clustering algorithm and the visualization approach that we use to present its results. We conclude the paper after a discussion of similar project and future work.

2 Interactive Visualization

In the process of user-adaptive clustering, the users interact directly with a community visualization that shows the community topology and the currently identified set of groups. Figure 1 shows a mapping of authors that have published in the UMAP conference series connected by co-authorship and similarity links[1].The visualization provides special affordances to guide the user through the interaction process [6].

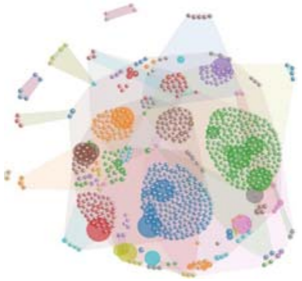

Fig. 1. The community visualization showing UMAP (User Modeling, Adaptation and Personalization) data

[1] This data was extracted from the DBLP [9] bibliography database, which created a 766 vertex and 8038 edge network. The similarity measure used is based on the Jaccard index [10].

The node saliencies include 1) nodes size based on degree centrality and 2) appended pie charts to highlight overlap and to what degree. The edges reflect co-authorship Jaccard similarity by increases the thickness of the line. The use of both convex hulls and inner-cluster distance minimized versus inter-cluster distance to outline and exacerbate group overlap and to differentiate the clusters. Vertices and text boxes provide a pointer cursor to make them known as selectable objects. To provide their individual views on community organization, the user has the option (and is encouraged) to select multiple vertices as a group and declare it as a clique that should belong to the same sub-community. The pictures below explain this process in detail.

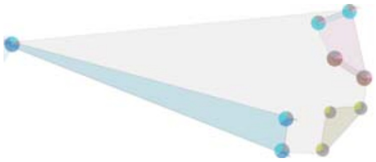

Fig. 2. Pie Charts Showcasing LPA correlation

Figure 2 shows a highlighted section of the graph that will be subject of user-defined groupings. The user selects all the nodes using control-clicks which is comparable to traditional multi-file select [7]. The nodes change their glyphs (circles to rhombus) and distinctly change to their most dominant group identity as highlighted in Figure 3. This provides two-dimensions of clustering aesthetics, both color (machine-derived clusters) and glyph changes (user-defined clusters).

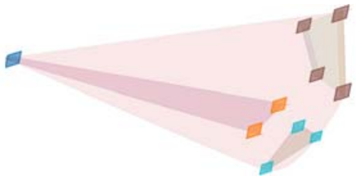

Fig. 3. User-Defined Clusters

After a set amount of seconds the new user-defined clique is automatically passed to our interactive clustering algorithm, presented in details in the next section, and after its completion the cluster assignments of the current dataset are updated in the visualization display. Finally, the user-defined group is differentiated by changing the opacity to be completely opaque, Figure 4. This process continues and the user-defined groups will be differentiated from one another by using new glyphs.

Fig. 4. Final Aesthetic to Highlight Completed Nodes[2]

3 Extending Label Propagation

To cluster the graph nodes, we applied an extended version of the Label Propagation Algorithm (LPA), which was proposed by Raghavan, Albert, and Kumara [8]. LPA iteratively determines the final cluster assignments. It initializes all vertices with a unique label, and then proceeds to update vertex labels by checking the labels of their neighbors. The most frequently occurring label among a vertex's neighbors is chosen as its new label. Ties are broken randomly. During all iteration, all vertices are (possibly) assigned a new label asynchronously. The process continues until it converges on a stable set of label assignments, which usually is within a few iterations.

We have extended the original LPA to allow it to be an active part in our interactive clustering visualization. We added to ability to run the algorithm on weighted graphs, changing the label picking criterion from the most frequent label of the neighbors to a version where the frequencies are modified by the edge weights. Furthermore, the labeling process was extended to allow for discovery of overlapping clusters. The calculated cluster label weights are further weighted by the proportions they appear in the label distribution of the adjacent vertices. A weight w_j for cluster label j is calculated using edge weights e_i of adjacent vertex i and cluster label proportion p_{ij} of adjacent vertex i for label j:

$$w_j = \sum_{i=1}^{n} e_i \frac{p_{ij}}{\sum_{j=1}^{c} p_{ij}}$$

The proportion weights p_{ij} for the new set of cluster labels of a vertex *i* are calculated after selecting the top *n* labels. The calculated weights of these *n* labels are normalized and stored with the new cluster labels of the vertex.

$$p_{ij} = \frac{w_{ij}}{\sum_{j=1}^{n} w_{ij}}$$

A very useful property of LPA is its near-linear time complexity [8]. Its swiftness makes it feasible to do the cluster calculations online. Specifically, we have the algorithm running in the background, waiting for updated information coming from

[2] This is a force-based graph and with new cluster assignments the subsequent topology will change. To keep consistent with explanation of the interactive process, we rotated the aforementioned areas to make it easier to follow.

the visualization component. To facilitate a fluid, adaptive cluster visualization experience, we have improved LPA to make this type of interaction feasible. The original LPA paper proposed an approach to seed a few vertices with cluster labels and leaving the rest unlabeled, allowing for clusters to form around those seeds. We have modified this approach by allowing vertices to be fixed. Once a vertex is fixed, it will no longer update its cluster label. This allows clusters to grow around vertices in a similar way to the original approach with the added benefit that we can choose to fix vertices in an already existing complete set of cluster assignments and rerun the algorithm on this cluster assignment to update it. Second, featured prominently in our visualization, is granting the user the ability to group vertices. Effectively, this allows grouped vertices to behave as a single vertex. When any of the vertices of the group updates its (set of) label(s), all others follow suit. Groups can also be fixed and if one of its members is a fixed vertex, the whole group becomes fixed.

4 Related Work

As mentioned above, mainstream software packages available for network data such as Gephi [3], Pajek [4] or Prefuse [11] are all limited to non-overlapping clustering approaches and do not allow the user the ability to put vertices into new groups without having to change meta-data about the vertex itself. Some research systems, however, explore both interactive clustering and overlapping communities that are distinguishing features of our approach.

Apolo [12] provides an approach in incorporating both user interactions and machine learning in large datasets. They accomplish this by building on a single vertex and as users provide a paper of interest to interface, the network-like visualization builds by providing cited work and visual clues based on number of citations and relevance. Building on this visualization, TourViz [13] divides sub-domains of interest (to the user) into convex hulls to help segment multiple topics of interest. In this context, our advancement takes into account the entire network structure, providing overall topology. Also, we provide not only the utilization of color changes (to distinguish group assignment), but also glyph distinction by user-defined clusters and opacity changes to discern vertices already selected and grouped.

5 Future Work

To continue this work we want to validate first the claim that our approach allows for easy understanding of community identity using convex hulls and adaptive clustering. To do this, we will need to provide a user-study of this mechanism and show that a mixture between user-defined and machine learned clustering can build an optimal and accurate model of the network topology and the user's mental model.

There is also a novel and yet strikingly obvious need to understand what it means to belong to multiple overlapping communities. Much work has been done in social capital that illustrates the strength of weak ties in bridging multiple communities [14]. We are interested in studying these networks more in depth to see if these vertices that fall between multiple communities can be defined more precisely using arguments from social capital. Gilbert [15] supports the claims made in this paper in regards to a spectrum to vertex-to-vertex variability and we believe that social capital and overlapping communes go hand in hand.

6 Conclusion

This paper presents an approach that allows to adapt a traditional cluster-based visualization of communities to user individual model of how the communities are organized. This kind of user-adapted visualization is possible because in a real world situation there are many alternative ways to segment users into heterogeneous groups since many users have affinity to more than one group. An interactive clustering and visualization approach presented in the paper allows the user to communicate their personal mental models of the communities to the clustering the algorithm itself and obtain community visualization picture that fits their expectations. To provide the user a fluid user-interface, both the visualization and the modified LPA was adapted to handle the size of the network and the interactions. Modifications were made to the visualization to showcase the communities in better quality and minimize edge crossing. The LPA was adjusted to allow for fixed vertices within the algorithm, allowing it to obtain an optimal solution in a relatively small amount of time. We believe that as networks are examined more in-depth, that platforms like this that take both visual information and user involvement can balance out both the human mental model of the network and the machine learning techniques used for efficiency.

7 Reference

1. Newman, M.E.J. and M. Girvan, *Finding and evaluating community structure in networks.* Physical review E, 2004. **69**(2): p. 026113.
2. Fortunato, S., *Community detection in graphs.* Physics Reports, 2010. **486**(3): p. 75-174.
3. Bastian, M., S. Heymann, and M. Jacomy, *Gephi: An open source software for exploring and manipulating networks.* 2009.
4. Batagelj, V. and A. Mrvar, *Pajek-program for large network analysis.* Connections, 1998. **21**(2): p. 47-57.
5. Borgatti, S.P., M.G. Everett, and L.C. Freeman, *Ucinet for Windows: Software for social network analysis.* 2002.
   networks visualization. in *Smart Graphics*. 2008: Springer.
6. Dieberger, A., et al., *Social navigation: techniques for building more usable systems.* interactions, 2000. **7**(6): p. 36-45.
7. Ritter, A. and S. Basu. *Learning to generalize for complex selection tasks*. in *Proceedings of the 14th international conference on Intelligent user interfaces*. 2009: ACM.
8. Raghavan, U.N., R. Albert, and S. Kumara, *Near linear time algorithm to detect community structures in large-scale networks.* Physical Review E, 2007. **76**(3): p. 036106.

9. Ley, M. *The DBLP computer science bibliography: Evolution, research issues, perspectives*. in *String Processing and Information Retrieval*. 2002: Springer.
10. Jaccard, P., *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. 1901: Impr. Corbaz.
11. Heer, J., S.K. Card, and J.A. Landay. *Prefuse: a toolkit for interactive information visualization*. in *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2005: ACM.
12. Chau, D.H., et al. *Apolo: making sense of large network data by combining rich user interaction and machine learning*. 2011: ACM.
13. Chau, D.H., et al. *TourViz: interactive visualization of connection pathways in large graphs*. in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2012: ACM.
14. Burt, R.S., *The social capital of structural holes.* The new economic sociology: Developments in an emerging field, 2002: p. 148-190.
15. Gilbert, E. and K. Karahalios. *Predicting tie strength with social media*. in *Proceedings of the 27th international conference on Human factors in computing systems*. 2009: ACM.