Interactively and Visually Exploring Tours of Marked Nodes in Large Graphs

Duen Horng Chau, Carnegie Mellon University, {dchau, lakoglu}@cs.cmu.edu
Leman Akoglu, Carnegie Mellon University, {dchau, lakoglu}@cs.cmu.edu
Jilles Vreeken, University of Antwerp, jilles.vreeken@ua.ac.be
Hanghang Tong, IBM T.J. Watson, htong@us.ibm.com
Christos Faloutsos, Carnegie Mellon University, christos@cs.cmu.edu

***Abstract*—We present TOURVIZ, a interactive system for visualizing and making sense of large network datasets. Given a set of user-specified nodes of interest, TOURVIZ integrates 1) novel algorithms to find the best subgraphs that succinctly connect these nodes; and 2) visualization and interaction features that help people explore such subgraphs.**

We will demonstrate TOURVIZ's usage and benefits using the DBLP co-authorship graph, which consists of 329K authors (nodes) and 1094K co-authorship relations (edges). TOURVIZ can work with any kinds of graphs. We will engage the audience to try our system and comment on its usability, usefulness, and how our system may help with their work and data analytics in their domains.

***Keywords*-Sensemaking, Networks, Connection subgraphs**

I. INTRODUCTION

Finding associations among a set of objects is an important problem in many domains ranging from biology (gene/protein interactions), security (criminal/terrorist interactions), immunology (patient interactions), and so on. Often, these objects are connected with certain types of relations within a large network. In such network settings, the task of finding associations among objects becomes *finding succinct connection pathways among those objects*. While these connection pathways reveal how the set of objects associate with one another in the network, the additional nodes (connectors) revealed on these pathways provide useful information about other nodes that might be of interest in understanding these associations.

We developed TOURVIZ (Figure 1), an interactive visualization system, for domain analysts to explore and understand the associations among a set of objects (i.e. nodes) in a network. An analyst selects a set of nodes that he is interested in within the network; these are the 'marked' nodes. Next, TOURVIZ uses an efficient algorithm to find succinct connection subgraphs among those marked nodes. Intuitively, 'close-by' (highly associated nodes) in the network can be connected by simple paths, while 'far away' (not-so-highly associated) nodes are hard to link together. Therefore, the algorithm returns possibly multiple connection subgraphs for multiple groups of 'close-by' nodes. TOURVIZ visualizes these groups and their associated connection subgraphs found by our algorithm. TOURVIZ also embodies interaction features for the analyst to update the nodes of interest on demand.

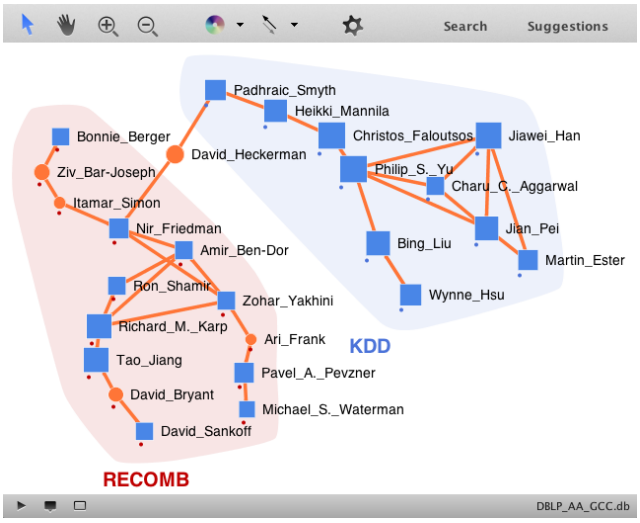

Figure 1. Screenshot of TOURVIZ showing a user exploring the connection subgraphs among a set of authors of interest (square nodes) in KDD and RECOMB (computational biology). Edges are co-authorship relations.

We summarize our main contributions as follows:

- Our TOURVIZ system helps users better understand how a set of nodes in a network are associated, using novel algorithms we developed [1] that finds succinct connection pathways among those nodes.
- TOURVIZ visualizes and reveals the pathways that connects users' nodes of interest, and enables the user to explore them in an interactively. Users can easily specify nodes of interest, and iteratively refine them.

II. DEMONSTRATING TOURVIZ

We will demonstrate TOURVIZ's usage, user interaction and algorithm through scenarios on understanding connections among researchers in various computer science fields in the DBLP[1] coauthorship graph, which contains about 329K authors (nodes) and 1094K coauthorship relations (edges).

Scenario. Here, we illustrate one example scenario, where our user uses TOURVIZ to explore and understand the connections between several researchers in KDD (data mining, machine learning) and RECOMB (computational biology). This scenario will touch upon major features of TOURVIZ.

[1]http://dblp.uni-trier.de/

Our user begins by searching for authors that he is familiar with, using TOURVIZ's search feature (Figure 2). Authors whose names contains the search text show up as a list. They can be sorted alphabetically, or by various metrics that TOURVIZ has pre-computed for the graph (the first time the graph is loaded), e.g., PageRank score, degree, etc. Once a match is found, our user drags the name into the visualization, which turns into a circle, its size proportional to its coauthor count (i.e., node degree).

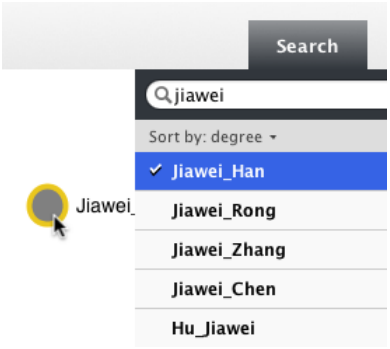

Figure 2. User searching for "jiawei". Matching authors shown in a list, sorted by the authors' coauthor counts (i.e., node *degrees* in the graph. Dragging the name of "Jiawei Han" into the visualization turns it into a circle; the name becomes the node label. Node size is scaled by degree. The yellow halo around the node indicates it is a match.

Our user proceeds to drag in more authors he is familiar with, in the domains of KDD and RECOMB. Although the user is familiar with these researchers' work, s/he does not recall how they have been collaborating. In particular, s/he is interested in using TOURVIZ to figure out the researchers who have been working at the intersection of these two domains.

To help keep track of the two groups of researchers, our user groups them into KDD and RECOMB using TOURVIZ's *grouping* feature (Figure 3). Each group is assigned a color, and its nodes enclosed by a convex hull.

These are all the nodes that our user wants to create a connection subgraph for (Figure 3). To mark these nodes as inputs, our user changes their shapes into squares (node colors do not matter). Next, they are given as input to TOURVIZ's algorithm [1] that finds the best subgraph. The algorithm finds a simple subgraph spanning all the input nodes that also has as few edges and as few additional intermediary nodes as possible.

Figure 4 shows the result of the algorithm—a tree whose edges are shown as thick orange lines, connecting the *marked* nodes with a few *intermediary* nodes (orange circles). Thin edges are other relations among all the nodes being displayed, but not part of the tree. Most interesting to our user is the discovery of David Heckerman; a prolific researcher who have been publishing with authors who frequent KDD and RECOMB. Additional discoveries include

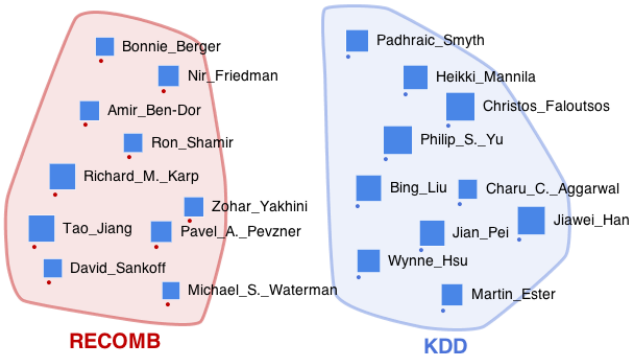

Figure 3. Our user has dragged multiple authors into the visualization, and grouped them into KDD and RECOMB using TOURVIZ's *grouping* feature. Each group is enclosed by a convex hull. Group names are provided by the user. Blue square nodes are the *marked* input nodes to TOURVIZ.

the intermediary nodes, that is the authors such as Ziv Bar-Joseph and Ari Frank, who are good connectors among the authors that our user is interested in observing the relations.

Our user can interactively add or remove nodes (marked or unmarked), and refresh the visualization with an updated connection subgraph by invoking the algorithm.

Engaging Our Audience. We will invite our audience to try out TOURVIZ with their own set of authors, and collect their feedback on TOURVIZ's usability and usefulness. Since we created TOURVIZ to be a general tool, we will be particularly interested in discussing with our audience how TOURVIZ may help them with research and data visualization in their domains.

III. SYSTEM DETAILS

The visualization component of TOURVIZ system is written in Java 1.7. built on top of the open-source JUNG network visualization library [3] and with design based on the APOLO [2] system. TOURVIZ stores the graph in a SQLite[2] embedded database, for its cross-platform portability. The graph database's schema was designed independently from the TOURVIZ system, so that different graphs that follow the schema can be readily used. TOURVIZ takes advantage of built-in features from SQLite, such as its full-text search capability to quickly locate nodes whose attribute values match users' search text (Figure 2).

When the user interactively specifies the set of marked nodes for which the connection subgraphs are to be found, their IDs are written to a temporary file. This file as well as the directory for the network data are input to our algorithm [1], implemented in Matlab 7.10, which outputs the best connection tree(s) as well as small subgraphs around the seed nodes (candidate graph). TOURVIZ shows the candidate graph and highlights (in bold) the edges that correspond to the best tree edges.

[2]www.sqlite.org

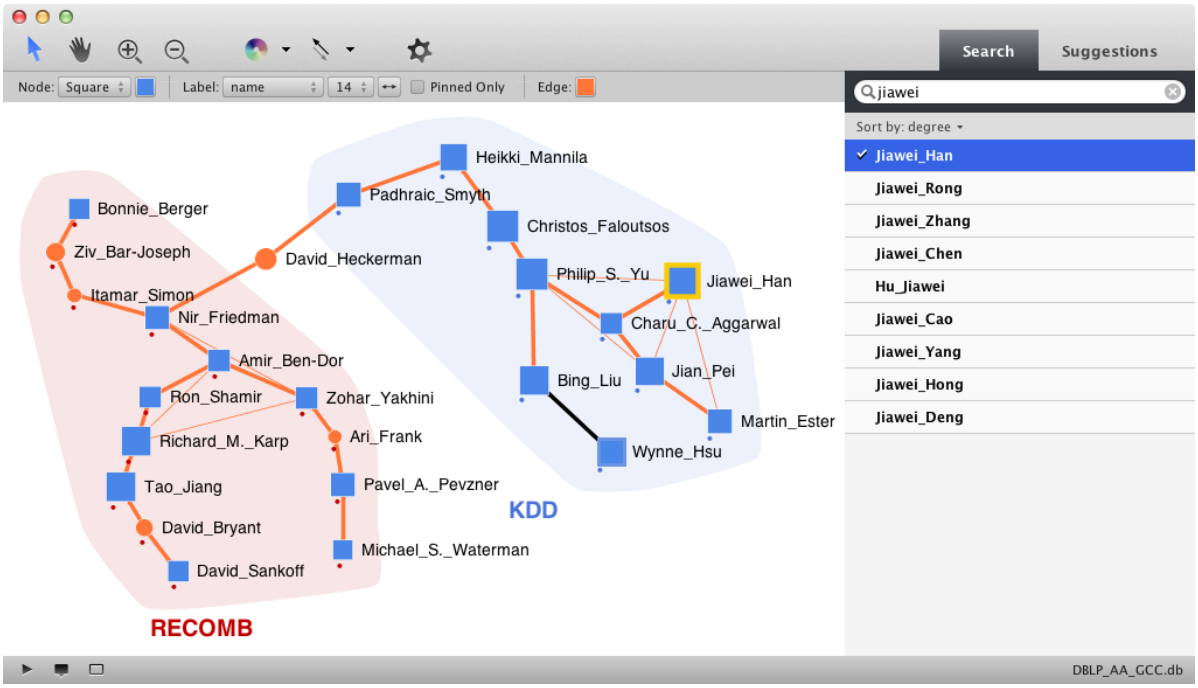

Figure 4. Screenshot of TOURVIZ showing our user exploring the connection subgraph among a set of authors of interest (blue, square nodes) in KDD and RECOMB (computational biology). Node shape is set using the first drop-down menu on the left; these nodes serve as input to the algorithm that finds the best tree (thick, orange edges) that connects them; other edges are shown as thin lines. Edges are co-authorship relations. Node size is scaled by the node's degree. TOURVIZ comes with useful visualization features: pan and zoom tool on the first row, color tools for nodes and edges, node label tool for specifying font size and node attribute to show (node "name" is chosen here). Our user can search for authors using the *Search Panel* on the right.

REFERENCES

[1] L. Akoglu, J. Vreeken, H. Tong, D. H. Chau, and C. Faloutsos. TourSum: Summarizing marked nodes in large graphs. In *Under review*, 2012.

[2] D. Chau, A. Kittur, J. I. Hong, and C. Faloutsos. Apolo: Making Sense of Large Network Data by Combining Rich User Interaction and Machine Learning. In *CHI*, 2011.

[3] J. O'Madadhain, D. Fisher, P. Smyth, S. White, and Y. Boey. Analysis and visualization of network data using JUNG. *Journal of Statistical Software*, 10:1–35, 2005.