RESEARCH ARTICLE OPEN ACCESS

Analysis and Visualization of Network Data Using JUNG

Pinaz Sayma
(M. Tech (CSE)-Student, Department of Computer Science, G.B.T.U., India)

ABSTRACT
The JUNG (Java Universal Network/Graph) Framework is a free, open-source software library that provides a common and extendible language for the manipulation, analysis, and visualization of data that can be represented as a graph or network. It is written in the Java programming language, allowing JUNG-based applications to make use of the extensive built-in capabilities of the Java Application Programming Interface (API), as well as those of other existing third-party Java libraries. In this paper, the JUNG Framework is described which is used to visualize data.
***Keywords* -**Data Analysis, Framework, Graph, Java, Network, Open-Source, Representation, Software Library, Visualization.

I. INTRODUCTION

A network data set typically consists of a set of entities and known relationships among these entities. For example, a social network data set could consist of a list of individuals and a list of pairwise binary relations indicating those pairs of individuals that are known to be friends. It is often convenient to formally represent such data as a graph, with vertices representing entities and edges representing their relationships. Examples of networks of broad interest include bibliographic citation networks (papers citing other papers), biological networks (e.g., proteins and their interactions), telecommunication networks, and the Web. Network data sets can also include additional information about both the entities and the rela- tionships. The challenges of working with network data motivate the creation and development of flexible software environments that are designed specifically for such data. And so JUNG framework is developed.

II. FEATURES OF JUNG

The major features of JUNG include the following:

1. Support for a variety of representations of entities and their relations, including directed and undirected graphs, multi-modal graphs (graphs which contain more than one type of vertex or edge), graphs with parallel edges (also known as multigraphs), and hypergraphs (which contain hyperedges, each of which may connect any number of vertices).
2. Mechanisms for annotating graphs, entities, and relations with metadata. These capabilities facilitate the creation of analytic tools for complex data sets that can examine the relations between entities, as well as the metadata attached to each entity and relation.
3. Implementations of a number of algorithms from graph theory, exploratory data analysis, social network analysis, and machine learning. These include routines for clustering, decomposition, optimization, random graph generation, statistical analysis, and calculation of network distances, flows, and ranking measures (centrality, PageRank, HITS, etc.)
4. A visualization framework that makes it easy to construct tools for the interactive exploration of network data. Users can choose among the provided layout and rendering algorithms, or use the framework to create their own custom algorithms.
5. Filtering mechanisms which extract subsets of a network; this allows users to focus their attention, or their algorithms, on specific portions of a network.

III. VISUALIZATION USING JUNG

JUNG provides mechanisms for laying out and rendering graphs. The current renderer im- plementations use the Java Swing API to display graphs, but they may be implemented using other toolkits (such as SWT). In general, a visualization requires one of each of the following:

1. A Layout, which takes a graph and determines the location at which each of its vertices will be drawn.
2. A (Swing) Component, which provides a "drawing area" upon which the data is ren- dered. JUNG provides a VisualizationViewer class for this purpose, which is an extension of the Swing JPanel class. A currently available experimental version of VisualizationViewer allows the user to create a "window" on the graph visualization, which can be used to magnify (zoom in on) portions of the graph, and

to select different areas for magnification (panning).

3. A Renderer, which takes the data provided by the Layout and paints the vertices and edges into the provided Component.

JUNG provides the GraphDraw class, which provides a framework for the interaction of these components by packaging the VisualizationViewer, the Renderer, and the Layout together. The default implementation fetches the location of each vertex from the Layout, paints each one with the Renderer inside the Swing Component, and paints each edge as a straight line between its vertices. Users may customize this behavior as desired; JUNG includes utilities and support classes that facilitate such customization. For instance, FadingVertexLayout provides a mechanism that can be used to create fading effects when vertices are filtered out and subsequently restored; this can be useful for highlighting ongoing changes, such as may occur during the temporal evolution of a social network.

The PluggableRenderer class is an implementation of Renderer that provides a number of ways for the user to customize the way in which the graph is rendered, including the vertex shape, size, color, and label, and the edge color, thickness, and label; each of these properties can be specified for each individual vertex or edge.

Since JUNG's data structures are completely separate from its visualization mechanisms, it is also possible to use other (Java or Java-compatible) visualization libraries to generate visualizations.

IV. RELATED WORK

JUNG was created out of a perceived need for a general, flexible, and powerful API for manipulating, analyzing, and visualizing graphs and networks. There exist numerous other packages and tools for visualizing and manipulating networks; Here is a comparison of JUNG with the following: UCINET (Borgatti, Everett, and Freeman (2004)), Pajek (Batagelj and Mrvar (2004)), R (R Development Core Team (2004)) with sna (Butts (2004)), and GFC (IBM Corporation (1999)).

UCINET and Pajek are stand-alone applications that each provide a number of tools for visualizing and analyzing networks. However, they cannot be conveniently addressed pro- grammatically by other applications, so they are not well-suited to process large numbers of graphs. Furthermore, they are applications rather than libraries, so users cannot write their own routines; this is particularly problematic for complex data sets (whose complexities may not have been anticipated by the application's designers) and for analysts that wish to use methods that are not provided as part of the application. Finally, they are closed-source projects, and executables are only available for the Windows environment.

R is a specialized programming language designed primarily for statistical computing and graphics. The sna package extends R in somewhat the same way that JUNG extends Java; like JUNG, sna provides a number of routines for social network analysis and visualization (including some 3-D visualization). R is well-suited for the rapid development of scripts and (in conjunction with sna) for on-the-fly network analysis, especially for analysis which requires sophisticated statistical tools. In addition, while it is not technically platform-independent, R is available on several popular platforms (so code written in R is reasonably portable), and is open-source.

GFC is a Java graph drawing-oriented API. It can only use Java's AWT/Swing mechanisms for rendering, contains few algorithms for network analysis, is no longer actively supported, and is not open-source.

V. FUTURE SCOPE

JUNG currently provides many of the tools and elements that are most commonly required for writing software that manipulates, analyzes, and visualizes network data sets. Future releases are planned to include the following features, several of which are currently under development. These features should significantly expand the set of available tools and enhance users' abilities to write robust code.

1. Expansion of the input and output options, including full implementation of database connectivity, ability to parse more complex Pajek and GraphML files, and support for other network file formats, such as the .dl and DOT formats.
2. Providing additional analysis tools: algorithms, statistical tests, parameter estimations, etc.
3. New visualization architecture: the current visualization system flexibly supports a broad set of visualizations, but is limited in certain respects (such as its ability to accommodate dynamic and evolving graphs, or to save and restore visual states). Continuing work on JUNG is investigating ways to adapt or replace the current system.
4. Creating a standard XML-based representation of JUNG's algorithms.

REFERENCES

[1] Adar E, Feinberg D (2005). "*GUESS: Graph Exploration System*." http://www.hpl.hp. com/research/idl/projects/graphs/.

[2] Alder G (2005). "*JGraph: The Java Graph Visualization Library*." http://www.jgraph. com.

[3] Apache Jakarta Project (2004). "*Commons Collections." http://jakarta.apache.org/ commons/c*ollections/.
[4] Batagelj V, Mrvar A (2004). "*Pajek: Program for Large Network Analysis*." http://vlado.fmf.uni-lj.si/pub/networks/pajek/.
[5] Battista GD, Eades P, Tamassia R, Tollis IG (1999). *Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall*.
[6] CERN (2004). "*The Colt Project*." http://dsd.lbl.gov/~hoschek/colt/.
[7] Eclipse Foundation (2001). "*SWT: The Standard Widget Toolkit*." http://www. Eclipse.org/articles/Article-SWT-Design-1/SWT-Design-1.html.
[8] Hoﬀ P, Raftery A, Handcock M (2002). "*Latent space approaches to social network analysis.*" Journal of the American Statistical Association, 97, 1090–1098.
[9] Open Source Technology Group (2005). "*SourceForge.net."* http://sourceforge.net.
[10] Penumarthy S, Mane KK, B¨orner K (2004). "*InfoVis Cyberinfrastructure Software Frame- work*." http://iv.slis.indiana.edu/sw/.
[11] R Development Core Team (2004). R: *A language and environment for statistical computing. R Foundation for Statistical Computing*, Vienna, Austria. 3-900051-07-0, URL http:// www.R-project.org.
[12] Smith MA, Fisher D (2005). "*Social Network Visualization of Usenet Newsgroups.*" To be presented at the International Sunbelt Social Network Conference.
[13] Sun Microsystems (2004). "*Creating a GUI with JFC/Swing.*" http://java.sun.com/docs/ books/tutorial/uiswing/.
[14] Wasserman S, Faust K (1994). *Social Network Analysis. Cambridge University Press, Cam- bridg*e.