A Process-Centric Data Mining and Visual Analytic Tool for Exploring Complex Social Networks

Denis Dimitrov
Georgetown University
Washington DC, USA
dd322@georgetown.edu

Lisa Singh
Georgetown University
Washington DC, USA
singh@cs.georgetown.edu

Janet Mann
Georgetown University
Washington, DC, USA
mannj2@georgetown.edu

ABSTRACT

Social scientists and observational scientists have a need to analyze complex network data sets. Examples of such exploratory tasks include: finding communities that exist in the data, comparing results from different graph mining algorithms, identifying regions of similarity or dissimilarity in the data sets, and highlighting nodes with important centrality properties. While many methods, algorithms, and visualizations exist, the capability to apply and combine them for ad-hoc visual exploration or as part of an analytic workflow process is still an open problem that needs to be addressed to help scientists, especially those without extensive programming knowledge. In this paper, we present Invenio-Workflow, a tool that supports exploratory analysis of network data by integrating workflow, querying, data mining, statistics, and visualization to enable scientific inquiry. Invenio-Workflow can be used to create custom exploration tasks, in addition to the standard task templates. After describing the features of the system, we illustrate its utility through several use cases based on networks from different domains.

1. INTRODUCTION

More and more social scientists and observational scientists have a need to understand and analyze complex network data sets. They need tools and methods that give them the opportunity to explore these data sets in an ad hoc manner or as part of an analytic workflow process. Kandel et al. [19] reiterate this need, saying that

> "little visualization research addresses discovery, wrangling or profiling challenges...Visual analytics tools that enable efficient application and assessment of these data mining routines could significantly speed up the analysis process."

Our work looks to help fill this gap by improving the capabilities of observational scientists to apply and assess data mining methods during data analysis.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IDEA'13, August 11th, 2013, Chicago, IL, USA.
Copyright 2013 ACM 978-1-4503-2329-1 ...$15.00.

This paper presents a tool (Invenio-Workflow) that supports exploratory analysis of network data by integrating workflow, querying, data mining, statistics, and visualization to enable scientific inquiry. Invenio-Workflow allows scientists to conduct a workflow process that adds data into the tool, queries the data, conducts a data mining task using one or more algorithms, and compares the results of the algorithms in a table or as an interactive graph visualization.

Some exploratory tasks that this tool supports include: finding communities/clusters that exist in the data using different algorithms, e.g. modularity and betweenness, predicting node labels using node/edge features and graph structure, identifying regions of similarity or dissimilarity between two data sets, querying and analyzing uncertain graphs, running network analysis using R, and finding and highlighting nodes with important centrality properties.

As an example, suppose a biologist wants to better understand group structure in an animal observational data set using Invenio-Workflow. This biologist may decide to create a workflow that compares the results of different clustering/community detection algorithms for the animal population's social network. Since the biologist may not be familiar with different community detection algorithms, she may want to compare the outputs of different methods to see which one matches intuition. Invenio-Workflow helps the biologist accomplish this by letting her setup a workflow that gets data from a file or a database, runs different clustering algorithms, and allows for visual exploration of the results. Figure 1 shows an example workflow for this scenario. While some standard task templates exist in Invenio-Workflow, a user can drag different widgets to create custom workflows to support exploration of data or existing analytic processes.

The contributions of this work include: 1) the development of a prototype process-centric, visual analytic tool; 2) a workflow process that includes visual, data mining, and graph query widgets for custom, exploratory analysis of network data; 3) integration of a graph query engine into the workflow process; and 4) an empirical demonstration of the utility of the proposed workflow design using a complex dolphin observation network and a citation network.

2. RELATED LITERATURE

2.1 Network/Graph Visual Analytic Tools

A number of excellent tools have been developed for exploring network data. Some of them are more general systems or toolkits [3, 4, 6, 7, 9, 12, 15, 18, 27, 26, 32, 14, 16, 34], while others are specialized for a specific task or analysis

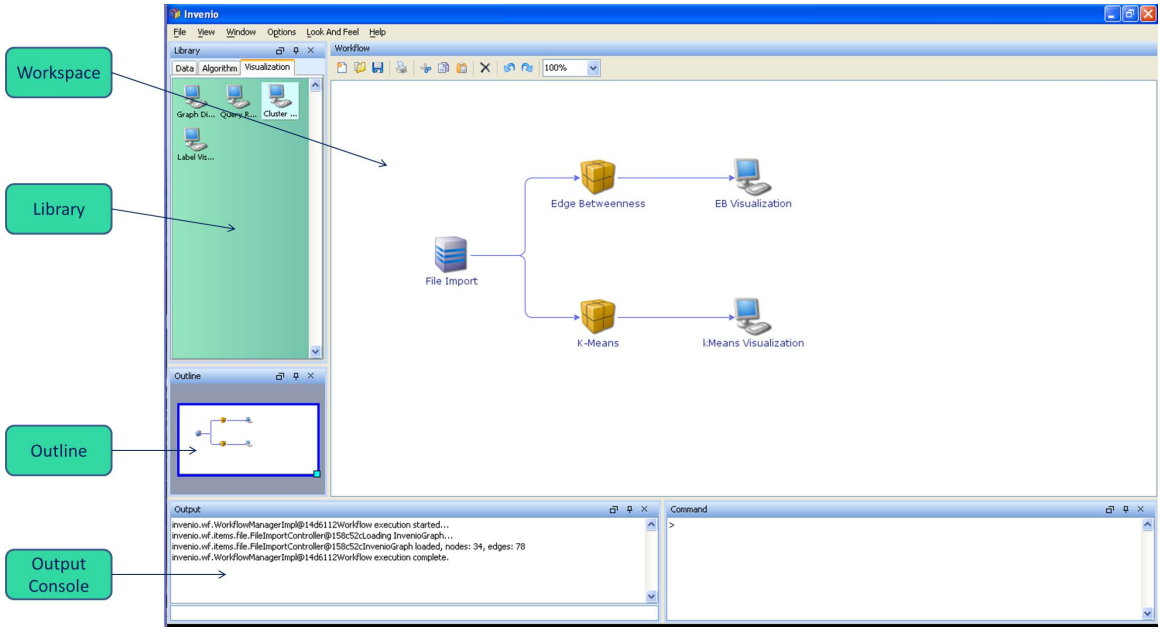


Figure 1: Clustering workflow in the context of Invenio-Workflow UI.

[8, 20, 22]. NodeXL [14] is an exploration tool that is integrated into Excel. Jung [27] is a toolkit containing a number of data mining algorithms. Prefuse [15] and Gephi [6] have more extensive visualization support. Guess [4] has a query language for manipulating graph visualization. While all of these tools have filtering, coloring, and interactive exploration capability, we view these tools as *data-centric* tools. In other words, they are stand-alone applications that focus on graph visualization and interactive manipulation of these graphs. In contrast, Invenio-Workflow is designed to be *process-centric*. Instead of focusing on manipulating and analyzing a single graph or network, our tool focuses on the entire process of data analysis and scientific inquiry, allowing the user to design and execute a workflow process that maps to a particular task of interest.

2.2 Workflow Tools

Orange [10], RapidMiner [24], and WEKA [13] are all examples of data mining / machine learning tools that consider the workflow process in their design. Orange is the most sophisticated in terms of workflow design, allowing users to create their own data mining workflow processes. It is the closest in design to Invenio-Workflow. Similar to Invenio-Workflow, widgets are used to setup an analytic workflow process. RapidMiner is the most sophisticated in terms of data mining support and visualizations. Its workflow support is more limited. The difference between these two tools and Invenio-Workflow is the goal of the tools. The developers of Orange have created a data mining tool that lets users setup different data mining processes. Invenio-Workflow is a scientific inquiry tool for graphs that lets users focus in on analyses specific to graphs of varying sizes. Its emphasis on graph data also distinguishes Invenio-Workflow from such mature, extensive software platforms as Amira [5] and Voreen [35], which incorporate the concept of dataflow for creating highly flexible, interactive visualizations of volumetric data.

Viztrails [31] is a tool that is designed with a similar purpose as our tool. It also combines databases, workflow systems, and visualization tools that are individually incomplete into an interactive scientific inquiry engine. The tool's emphasis use-case is provenance. In contrast, our tool incorporates extensive graph mining and statistics components. The graph mining includes a number of custom algorithms for community detection, bias analysis, etc. For statistical support, Invenio-Workflow incorporates *R* [30]. Many of these analyzes are not supported by Viztrail.

3. TOOL DESCRIPTION

Invenio-Workflow provides an interactive, process-centric environment for graph exploration. Users visually define a process choosing from the available widgets and connecting them to designate the desired data flow. Each widget represents a logically separate piece of functionality, such as loading a graph from a file or executing a specific algorithm.

The Invenio-Workflow interface, shown in Figure 1, consists of several panels. The *Workspace* serves as an editor for constructing workflows, i.e. inserting, connecting, and re-arranging widgets. In addition, it has capabilities for zooming, copying and pasting widgets, and saving workflows for later loading and reuse.

Desired widgets are inserted by dragging from the *Library* panel into the *Workspace*. The *Library* contains several categories of widgets. Data import widgets are used to load data from different sources and formats. Algorithm widgets encompass data processing functionality: specific clustering algorithms, node labeling algorithms, and graph query execution are the main ones in our tool. Visual widgets include a general graph visualization widget, as well as several visualizations best suited for analyzing results of specific algorithms.

The *Outline* panel facilitates navigating large workflows. Sliding the workflow focus window over the workflow outline and adjusting its size brings the corresponding workflow area

into the *Workspace* editor. The *Output* panel is a logging console, to which Invenio-Workflow and its components post messages, warnings, and errors.

We will demonstrate the Invenio-Workflow interface using the previously mentioned motivational example of clustering / community detection algorithms in the well-known karate social network [36]. In this network data set nodes correspond to members of a karate club at a US university and edges represent friendships. Sociologists have used this network for a number of different studies, including ones related to group structure and models of conflict. Sociologists may be interested in seeing which algorithm captures the expected group structure most accurately, or in cases where they do not know the group structure, compare the outputs of the different clustering algorithms to identify group structure.

The *Workspace* in Figure 1 shows the workflow created for our example task - comparing clustering algorithms and visually exploring the results. The File Import widget loads the data set graph from a text file in one of the supported formats. This graph is forwarded to two clustering algorithms: Edge Betweenness Clusterer and K-Means Clusterer. The output of each algorithm is connected to an interactive Cluster Visualizer. This visualizer allows the researcher to see which nodes are member of different clusters, as well as compare the similarities and differences between the two clusterings.

Before a workflow can be executed, some widgets may need to be configured. For example, the File Import widget needs to be supplied with the location of the data set file(s), by opening up its configuration editor. Some clustering algorithms also need parameters specified. Figure 2 shows the configuration panel for the Edge Betweenness Clustering algorithm. After configuring the widgets and executing the workflow, the results at different steps can be observed by opening up the result view of the corresponding widget. Depending on its functionality, a widget may offer a configuration editor, a result view, both, or possibly neither. They open up in separate frames, which the user can dock or float anywhere over the workspace. They can remain open, as long as the widget is present in the workflow, so that the user can observe the configuration and results of any number of widgets simultaneously.

Executing the demonstrated workflow, we obtain the Cluster Visualization result views in Figure 3a and Figure 3b. The information panel provides the user with basic statistics about the generated clusters, such as the number of clusters and the minimum, maximum, and average number of nodes in a cluster. Selecting one or more clusters from the list highlights the corresponding nodes in the graph view. There is also an option to hide the remaining nodes. By viewing the resulting clusterings from the two algorithms, a researcher can analyze the clusters that are similar in both algorithms, and those that differ.

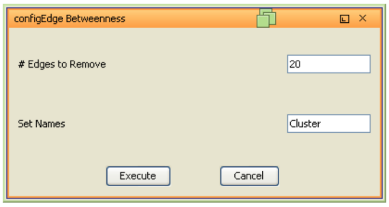


Figure 2: Configuration Panel for Edge Betweenness Cluster widget.

The process-centric approach has several advantages. It provides for ease and flexibility for visually defining, changing, and executing analytical tasks as opposed to writing code or using monolithic, pre-defined tools. Saving and loading workflows makes it possible to repeat the analysis in the future, possibly on a different data set, as well as to share it with other users.

Our implementation is an initial proof of concept, containing widgets that handle only a small subset of possible graph manipulation and visualization tasks. Therefore, one of our design priorities was an open architecture that allowed for ease when adding new widgets. In the straightforward case, a new widget is added by implementing a simple interface and registering the widget with Invenio-Workflow. Additionally, during deployment the widget may be associated with a configuration editor and / or result view. It may declare the expected number and type of input data. For more complex validation, the widget may specify validators to be invoked by the framework at runtime.

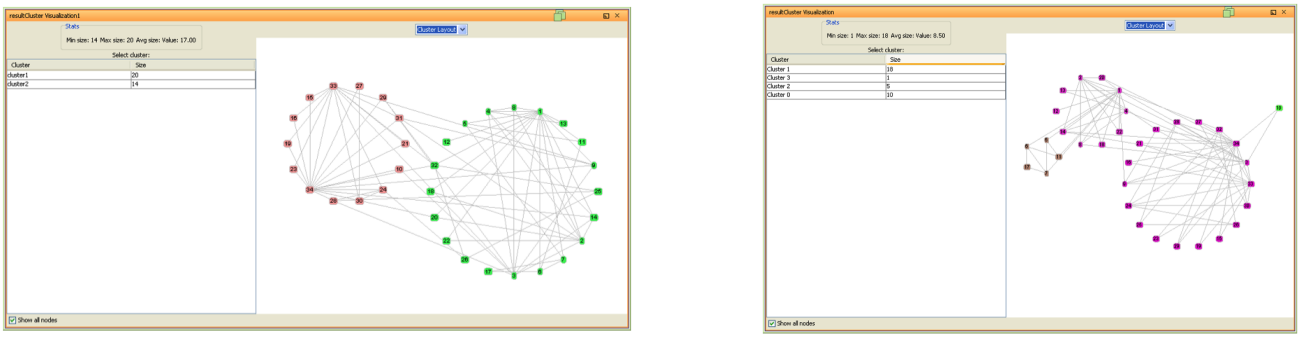
Invenio-Workflow is written in Java. For most flexibility, the workflow processing engine and the individual widgets are a custom implementation. However, Invenio-Workflow relies on several toolkits to support various other aspects of its functionality. Workflow diagrams are based on the open-source JGraph framework [1]. JIDE Docking Framework [2] is used for dockable window support. Graph visualizations in the corresponding widgets build upon the Prefuse visualization toolkit [15, 28]. The uncertain graph data model is implemented as an extension of the JUNG graph data model [17, 27]. Node labeling algorithm implementations are provided by GIA [25]. R [29, 30] is integrated for statistical analysis. Graph query processing is delegated to a query engine that was developed by our group.

4. USE CASES

This section presents use cases based on two real-world data sets, introducing several widgets and demonstrating how they can be customized and combined for accomplishing the desired analytical tasks. For these use cases, we use a personal computer with dual core 2.9 GhZ processor and 8 GB of memory. The graphs analyzed both fit into main memory.

4.1 Dolphin observation network use case

This use case considers a dolphin social network based on approximately 30 years of study of a dolphin population in Shark Bay, Australia [23]. The data set includes demographic data about approximately 800 dolphins, represented as graph nodes and social interactions between these dolphins, captured as approximately 29,000 edges. A researcher observing a particular animal may be uncertain about its identification, features, or behavior. This uncertainty can be expressed as existence probabilities between 0 and 1, associated with nodes and/or edges and as categorical uncertain attributes, representing discrete probability distributions over the set of possible attribute values. Dolphin vertices in the data set include certain attributes (id, confidence, dolphin_name, birth_date) and uncertain attributes (sex_code, location, mortality_status_code), while edges have attributes (id, confidence).



(a) K-Means.

(b) Edge Betweenness.

Figure 3: Result view of 2 different Cluster Visualization Widgets.

A scientist can obtain a general idea about the size, density, and connectivity of the network by creating a very simple workflow that loads the graph and feeds it into a standard graph visualization widget. The latter displays the graph along with basic statistics, such as number of vertices / edges and average degree. To give the users the capability for in-depth exploration and comparison of uncertain graphs, we have designed a prototype SQL-like query language that combines elements of relational, uncertain, and graph databases [11]. In this work, we have incorporated our query engine implementation into a widget that allows users to write their own ad-hoc uncertain graph comparison queries.

Our team met with observational scientists who work with the dolphin population and developed a list of typical queries that they would like the capability to issue when analyzing this dolphin social network and its inherent uncertainty, including:

- Selecting the number of associates and sex composition of associates for male and female dolphins, respectively, using the most probable value of the *sex_code* attribute.
- Visualizing the union, intersection, difference, and bi-directional difference between the ego-networks of a particular dolphin during two different years, where the confidence of relationship existence is above a specified threshold.
- Finding the common associates (friends) of two specific dolphins with a relationship confidence above a certain threshold.
- Calculating a measure of structural and semantic similarity between ego-networks of two particular dolphins.

These and many other queries can be expressed in the proposed language and executed using the Query widget. Because the query result represents a relation whose tuples may contain graphs, vertices, edges, attributes, and / or primitive types, depending on the query, the Query Result widget helps the user to visually explore the result. Figure 4 shows the result of executing a query that returns union, intersection, and difference between ego-networks of a particular dolphin (JOY) during years 2010 and 2009, respectively[1]. The bottom panel contains the executed query. The table on the left is the resulting relation, in this example consisting of a single tuple. As specified in the query, the tuple's columns contain the ego-networks for each year, and their union, intersection, and difference. The main panel visualizes the value, selected in the result table: in this case, the union of ego-networks between the two years, which represents a graph of dolphins that were observed together with JOY during at least one of the years. By selecting the "intersection" and "difference" columns, the researcher can visualize JOY's repeat friends and new friends, respectively, and discover that the two sets, although disjoint, are approximately the same size (due to space limitations, these results are not shown). Selecting a vertex in columns n1 (node 1 attributes) and n2 (node 2 attributes) changes the display to present the vertex attributes instead of the graph visualization.

By connecting several instances of the Query Result widget to the output of the same Query widget instance, the researcher can manipulate them independently to obtain simultaneous different views of the same query result. For example, it may be helpful to display the ego-network for each year (columns ego1 and ego2) and visually compare them side-by-side using union, intersection, and difference operators.

While we do not have space to discuss our query language or present the related queries in more detail, when executing these queries we made several observations. We can visually observe that dolphins who are most probably males are seen together more often than any of the other combinations of sex. Furthermore, a simple query that calculates the average number of associates for male and female dolphins, confirms that male dolphins are more social on average: 51.2 associates compared to 32.6 for female dolphins. Using one of our similarity operators, we identified potentially similar ego-networks to JOY's ego-network. As expected, ego-networks from dolphins who are observed in the same area as JOY had a higher similarity score, since dolphins are likely to have similar associates if they have the same primary location.

4.2 Citation network use case

The second use case is based on the Cora document citation data set [21]. It contains 2708 nodes, representing machine learning papers, and 5429 edges representing citations to the papers. Each publication is described by a 0/1-valued word vector indicating the absence/presence of

[1]the order is important for the difference operator

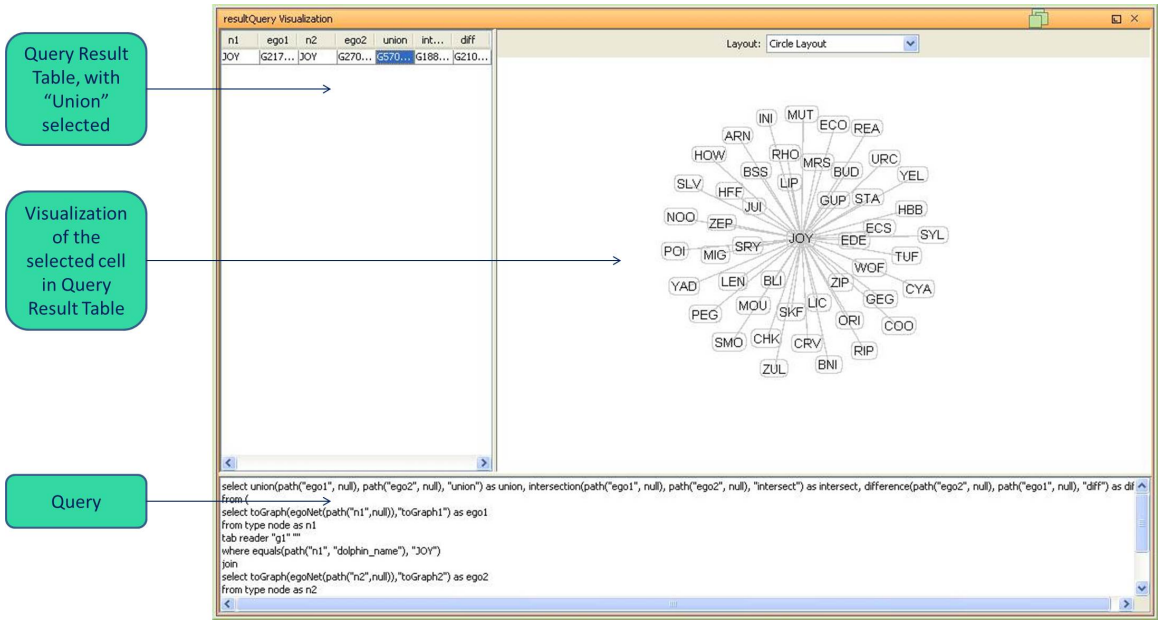


Figure 4: Visualizing query results.

the corresponding word from the dictionary of 1433 unique words. The goal of this use case is to use some or all of the network structure information and the word vectors of the papers to determine the topic of the paper. Each paper can be classified into one of seven possible topics ("Case Based", "Genetic Algorithms", "Neural Networks", "Probabilistic Methods", "Reinforcement Learning", "Rule Learning", "Theory").

More specifically, this use case involves examining and comparing the output of two different node labeling algorithms, which use a partially observed citation data set to predict the probability distribution of the label, i.e. the topic attribute. The workflow in Figure 5a shows two of the several algorithms offered by Invenio-Workflow, where the choice and configuration of the particular algorithm is performed within the Node Label widget. The Majority Rule algorithm simply calculates the label distribution using the labels of the adjacent vertices. The Naïve Bayes classifier, on the other hand, disregards the graph structure and predicts the document topic based on the other attributes, in this case occurrence of words in the paper. We ran the algorithms using a 2-fold training / testing split. Each algorithm partitions the original data into two sets, trains on each set to produce predictions for the other, and outputs a copy of the graph with the predicted probability distribution of the paper topic for each node (paper) in the graph.

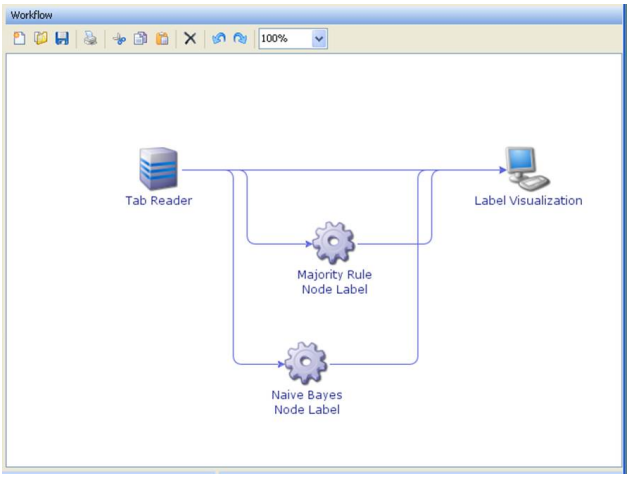
Comparing and contrasting these uncertain graphs to each other or to a ground-truth graph allows researchers to analyze the performance of different node labeling algorithms, experiment with a single algorithm under different assumptions, and examine the graph data set, by highlighting parts of data where algorithms disagree in their predictions or perform poorly. To that end, we created the Node Label Visualization widget. It takes as input the two predicted uncertain graphs, as well as the ground truth graph.

The widget's result view is shown in Figure 6. The basic visual analytic concept is borrowed from the G-Pare visual analytic tool [2] [33]. The statistics pane determines that the simple Majority Rule classifier has better overall accuracy. The table above it provides a side-by-side comparison at vertex granularity. Its columns show the vertex identifier and the most probable label from the ground truth graph and each of the two predicted graphs, respectively. The last three columns show the color-coded histogram representing the probability distribution over all possible labels - again, based on the ground truth and the two node labeling algorithms. The height of the bar corresponds to the probability of the particular value being the actual label. Hovering the mouse over a distribution cell brings up a tooltip with the exact probability for each of the possible label values. Choosing a column highlights the corresponding vertex in the graph view on the right, allowing users to visually identify the node and its neighbors.

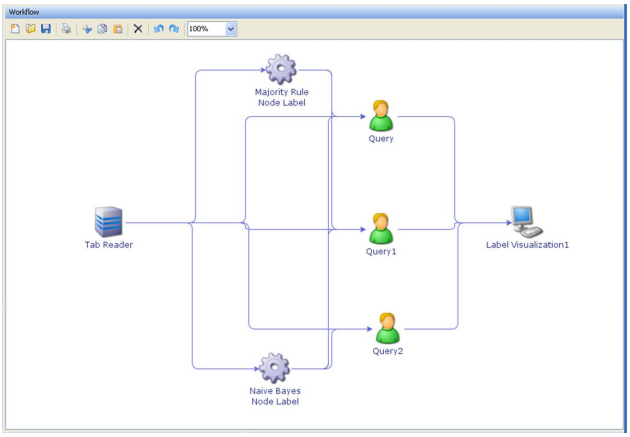
Unlike the detail table, which focuses on examining predictions for each node, the table in the lower left corner provides a higher-level view, showing the areas where the node labeling algorithms agree and disagree in their predictions. As described in [33], the table represents a confusion matrix between the pair of graphs, selected in the combo box, with the option to compare each graph against the ground truth or between each other. Each cell in the table contains the count of nodes with the corresponding combination of most probable label values between the two chosen graphs. Therefore, the cells along the main diagonal represent the vertices, whose most probable labels match between the graphs.

Through heatmap coloring ranging from green through yellow to red, the confusion matrix emphasizes the relative frequency of nodes in each cell. Hovering the mouse over a particular cell brings up the percentage of its node count relative to the total number of nodes in the graph. Furthermore, the confusion matrix supports selection and filtering. Choosing a cell eliminates all the nodes belonging to other

[2]G-Pare is a tool focused on analyzing and comparing machine learning algorithms related to the node labeling task.



(a) Workflow for comparing node labeling algorithm results.



(b) Workflow that includes queries for filtering node labeling algorithm results.

Figure 5: Node labeling workflows.

cells from the detail table. It also highlights the corresponding nodes in the graph view on the right. There is an option to hide the nodes that are not selected in either the confusion matrix or the detail view. For example, the graph view in figure 6 shows the graph subset that includes only "Neural Network" papers misclassified as "Theory" by the Naïve Bayes algorithm.

The graph view, displaying the whole graph or its selected subset, assists the user in visually identifying the differences in predicted labels between the chosen pair of graphs. When the two graphs have the same most probable value, the nodes are colored in different intensity of green through blue, depending on the difference in probability between the graphs. Likewise, the nodes for which the label does not match are colored in shades of yellow through red.

Applying these capabilities to the Cora data set, we can see that for both algorithms, the counts along the main diagonal of the confusion matrix are significantly higher than in the remaining cells, confirming the relatively high accuracy reported in the statistics panel. In particular, "Neural Networks" stands out as the category that occurs by far the most both in the ground truth and in the predicted graphs. It is also the category, in which the Naïve Bayes classifier under performs in comparison with the other categories, in which the counts between the two models are relatively close. In this category, the algorithm misclassified a higher number of papers as either "Probabilistic Methods" or "Theory". Visually examining the detail table and then filtering for nodes misclassified as "Theory", we can conclude that most Naïve Bayes predictions are inaccurate with high probability. The Majority Rule classifier, while being less certain in many cases, is able to suggest the correct topic.

Inserting a query into the node labeling workflow is a flexible way to complement the basic filtering capabilities embedded in the Node Label visualization widget. Such enhanced workflow (Figure 5b) lets the analyst select and concentrate on some areas of particular interest. For example, we can write a query selecting the subgraph that includes only the vertices, for which both models give wrong predictions with high probability (greater than 0.75) and the edges between these vertices.

The Query widget has the capability of selecting a particular cell from the query result relation and sending the selected object downstream to the connected widgets, instead of sending the whole table. Supplying the Node Label visualization with the desired subgraph of papers misclassified with high confidence yields the visualization in Figure 7.

The statistics panel shows the user that the subset under examination is relatively small. With only 89 nodes, it does not provide enough information to draw reliable conclusions about the models on bigger scale, only to make observations and suggest directions for further examination. One of these observations is that, with only 15 edges between the 89 vertices, the papers under consideration are mostly unrelated to each other. There are several exceptions, consisting of 2 or 3 papers, as observed in the graph view. As expected, when the accuracy is 0, there are no entries along the main diagonal, and all nodes in the graph are red, indicating strong label mismatch.

Examining the three confusion matrices leads to several observations (Figure 8). Model 1 misclassified a number of Neural Networks papers as Probabilistic Methods papers and vice versa. This is mostly consistent with model 2 predictions, but in addition, model 2 also assigned Theory label to a number of Neural Networks papers. Furthermore, comparing model 1 vs model 2, the user can see that the majority of entries are along the main diagonal (especially in Theory category), and most nodes in the graph view are colored blue. This leads to the conclusion that whenever both models were wrong with high probability, they made the same predictions, showing the common limitation that both classifiers have or the possible noise in the data set.

Clicking along the main diagonal between the two models and observing the changes in the detail table provides a different perspective. The user may notice that in most cases, there is no obvious correlation between the actual topic and the topic simultaneously chosen by both models (Figure 9). The exception, however, is the case shown in Figure 10, where the majority of papers collectively misclassified as Probabilistic Methods are in fact Neural Networks, and vice versa (not shown). These findings reinforce the previously made observations.

This query represents a single example of examining the results of two node labeling algorithms over a particular subset of interest. Additional queries can be introduced into the

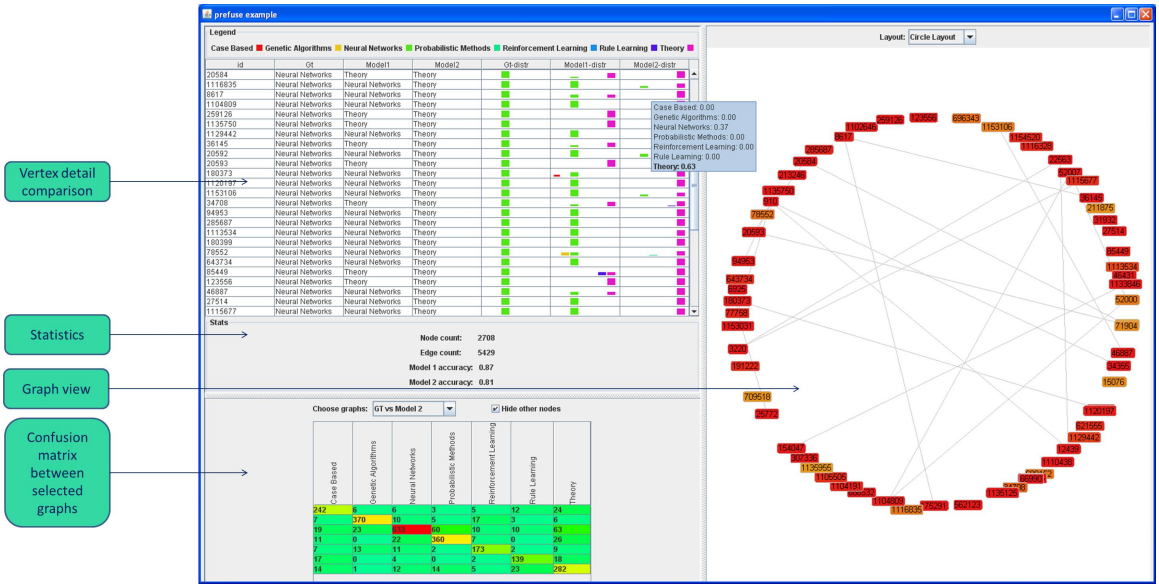


Figure 6: Visualization of node labeling algorithm results.

workflow to further investigate reasons for the wrong predictions. Or, by creating similar queries or modifying these queries in-place, researchers can evaluate other subsets.

5. CONCLUSIONS

In this paper, we have demonstrated how an analyst may use Invenio-Workflow to create workflows with widgets for importing data, executing algorithms, running queries, and interacting with visualizations. Our prototype has focused on three tasks, clustering, node labeling, and graph comparison. We have shown that the widgets associated with these tasks can be synergetically combined in different ways that solve a variety of analytical tasks beyond the capabilities of each single widget. Furthermore, building / executing workflows, interactively analysing their results, and modifying / re-running the workflow in an iterative, ad-hoc manner is a valuable capability for analysts dealing with complex network data, particularly observational scientists. By bringing together elements from areas that include workflow processing, uncertain graph queries, data mining, and graph visualization, we believe that we have created a unique tool with abilities to examine, analyze, and visualize a wide range of graph data from different domains. As future work, we hope to increase the number of data mining and other exploratory tasks supported by the tool, optimize performance for graphs that do not fit into main memory, and develop more sophisticated widgets related to time-evolving networks and information diffusion.

Acknowledgments

This work was supported in part by the National Science Foundation Grant Nbrs. 0941487 and 0937070, and the Office of Naval Research Grant Nbr. 10230702.

6. REFERENCES

[1] Jgraph - open source (bsd) java graph visualization and layout component. http://www.jgraph.com/jgraph.html.

[2] The jide docking framework - a very powerful yet easy-to-use dockable window solution. http://www.jidesoft.com/products/dock.htm.

[3] J. Abello, F. van Ham, and N. Krishnan. ASK-graphview: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):669–676, 2006.

[4] E. Adar. Guess: a language and interface for graph exploration. In *International Conference on Human Factors in Computing Systems*, 2006.

[5] Amira. Software platform for 3d and 4d data visualization, processing, and analysis. http://www.amira.com/.

[6] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An open source software for exploring and manipulating networks. In *International AAAI Conference on Weblogs and Social Media*, 2009.

[7] V. Batagelj and A. Mrvar. Pajek-analysis and visualization of large networks. In P. Mutzel, M. Junger, and S. Leipert, editors, *Graph Drawing*, volume 2265 of *Lecture Notes in Computer Science*. Springer, 2002.

[8] M. Bilgic, L. Licamele, L. Getoor, and B. Shneiderman. D-dupe: An interactive tool for entity resolution in social networks. In *IEEE Symposium on Visual Analytics Science and Technology*, 2006.

[9] U. Brandes and D. Wagner. Visone - analysis and visualization of social networks. In *Graph Drawing Software*, 2003.

[10] J. Demšar, B. Zupan, G. Leban, and T. Curk. Orange: from experimental machine learning to interactive data mining. In *European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2004.

[11] D. Dimitrov, L. Singh, and J. Mann. Comparison queries for uncertain graphs. In *(to appear) 24th*

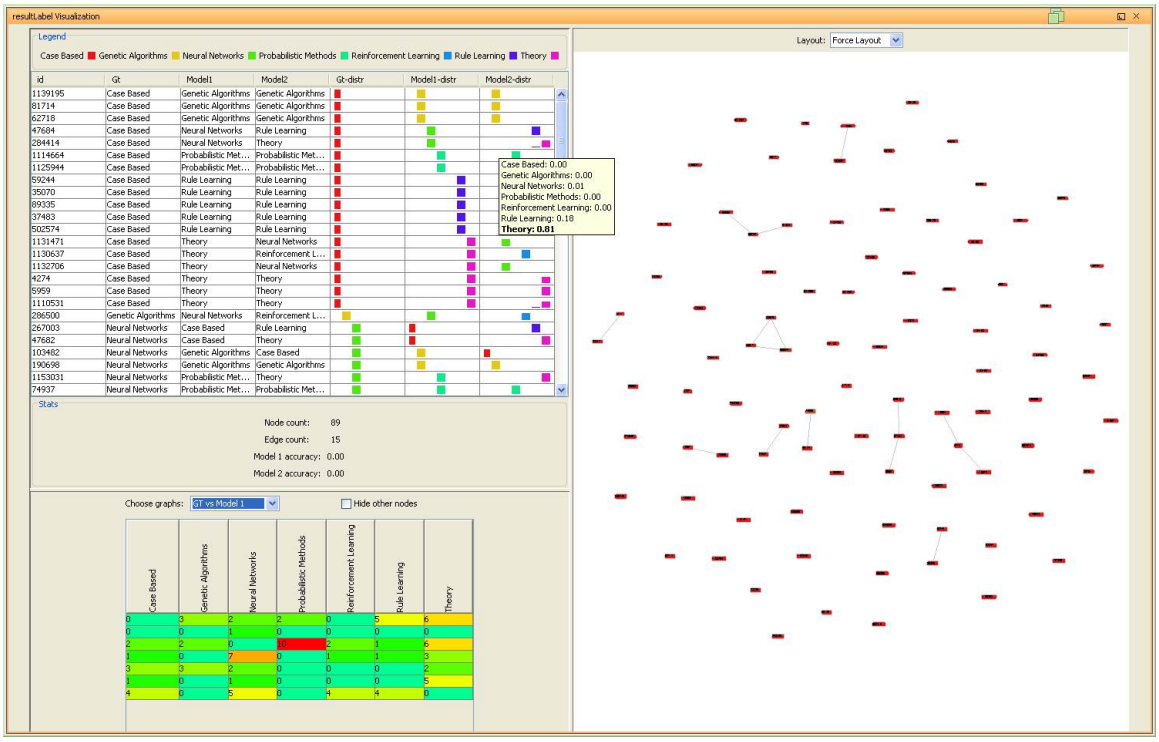


Figure 7: Visualization of query-filtered node labeling algorithm results.

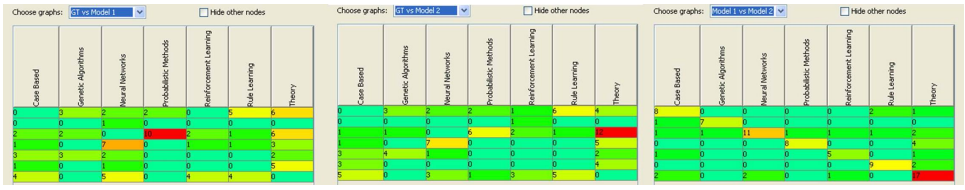


Figure 8: Confusion matrices for query-filtered nodes.

International Conference on Database and Expert Systems Applications, DEXA, 2013.

[12] M. Freire, C. Plaisant, B. Shneiderman, and J. Golbeck. ManyNets: an interface for multiple network analysis and visualization. In *International Conference on Human Factors in Computing Systems*, 2010.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11:10–18, 2009.

[14] D. Hansen, B. Shneiderman, and M. A. Smith. *Analyzing Social Media Networks with NodeXL: Insights from a Connected World*. Morgan Kaufmann Publishers, 2011.

[15] J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *International Conference on Human Factors in Computing Systems*, 2005.

[16] N. Henry, J. Fekete, and M. J. McGuffin. Nodetrix: A hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 13:1302–1309, 2007.

[17] JUNG. Java universal network/graph framework. http://jung.sourceforge.net/.

[18] I. Jusufi, Y. Dingjie, and A. Kerren. The network lens: Interactive exploration of multivariate networks using visual filtering. In *Conference on Information Visualisation*, 2010.

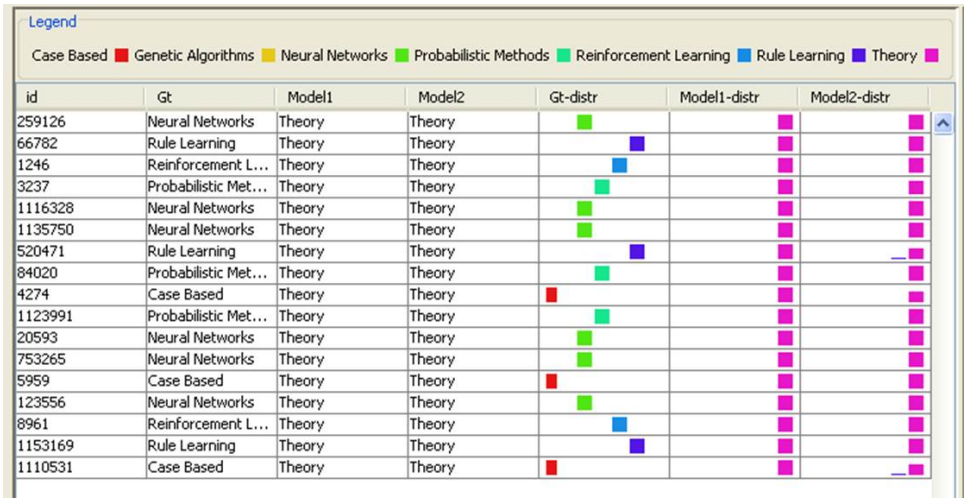
[19] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Enterprise data analysis and visualization: An interview study. In *IEEE Visual Analytics Science & Technology (VAST)*, 2012.

[20] H. Kang, L. Getoor, and L. Singh. C-group: A visual analytic tool for pairwise analysis of dynamic group membership. In *IEEE Symposium on Visual Analytics Science and Technology*, 2007.

[21] LINQS. Machine learning research group @ umd. Available from http://www.cs.umd.edu/ sen/lbc-proj/LBC.html.

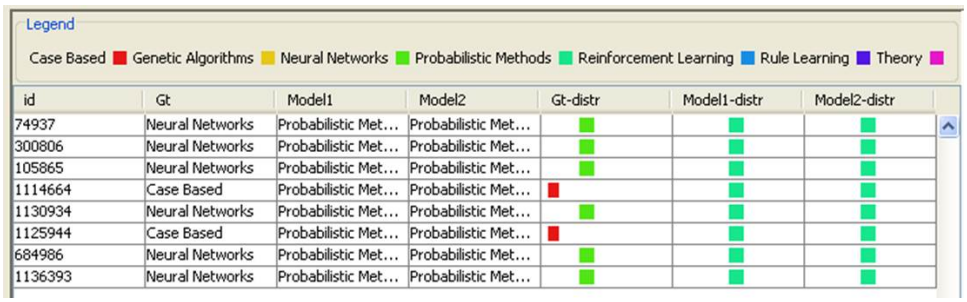
[22] Z. Liu, B. Lee, S. Kandula, and R. Mahajan. Netclinic: Interactive visualization to enhance automated fault diagnosis in enterprise networks. In *IEEE Symposium on Visual Analytics Science and Technology*, 2010.

[23] J. Mann and S. B. R. Team. Shark bay dolphin project. http://www.monkeymiadolphins.org, 2011.



id	Gt	Model1	Model2	Gt-distr	Model1-distr	Model2-distr
259126	Neural Networks	Theory	Theory			
66782	Rule Learning	Theory	Theory			
1246	Reinforcement L...	Theory	Theory			
3237	Probabilistic Met...	Theory	Theory			
1116328	Neural Networks	Theory	Theory			
1135750	Neural Networks	Theory	Theory			
520471	Rule Learning	Theory	Theory			
84020	Probabilistic Met...	Theory	Theory			
4274	Case Based	Theory	Theory			
1123991	Probabilistic Met...	Theory	Theory			
20593	Neural Networks	Theory	Theory			
753265	Neural Networks	Theory	Theory			
5959	Case Based	Theory	Theory			
123556	Neural Networks	Theory	Theory			
8961	Reinforcement L...	Theory	Theory			
1153169	Rule Learning	Theory	Theory			
1110531	Case Based	Theory	Theory			

Figure 9: Detail Table for query-filtered nodes misclassifed as Theory by both algorithms.



id	Gt	Model1	Model2	Gt-distr	Model1-distr	Model2-distr
74937	Neural Networks	Probabilistic Met...	Probabilistic Met...			
300806	Neural Networks	Probabilistic Met...	Probabilistic Met...			
105865	Neural Networks	Probabilistic Met...	Probabilistic Met...			
1114664	Case Based	Probabilistic Met...	Probabilistic Met...			
1130934	Neural Networks	Probabilistic Met...	Probabilistic Met...			
1125944	Case Based	Probabilistic Met...	Probabilistic Met...			
684986	Neural Networks	Probabilistic Met...	Probabilistic Met...			
1136393	Neural Networks	Probabilistic Met...	Probabilistic Met...			

Figure 10: Detail Table for query-filtered nodes misclassifed as Probabilistic Methods by both algorithms.

[24] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. Yale: Rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, New York, NY, USA, August 2006. ACM.

[25] G. Namata. Gaia (graph alignment, identification and analysis), a software library and tool for analyzing and running machine learning algorithms over graph data. https://github.com/linqs/GAIA.

[26] NetMiner. Netminer - social network analysis software. Available from http://www.netminer.com.

[27] 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.

[28] Prefuse. The prefuse visualization toolkit. http://prefuse.org.

[29] R. Software environment for statistical computing and graphics. http://www.r-project.org/.

[30] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012.

[31] E. Santos, D. Koop, H. T. Vo, E. W. Anderson, J. Freire, and C. Silva. Using workflow medleys to streamline exploratory tasks. In *Proceedings of the 21st International Conference on Scientific and Statistical Database Management*, SSDBM, 2009.

[32] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, 2003.

[33] H. Sharara, A. Sopan, G. Namata, L. Getoor, and L. Singh. G-pare: A visual analytic tool for comparative analysis of uncertain graphs. In *IEEE VAST*, pages 61–70, 2011.

[34] J. Stasko, C. Gorg, and Z. Liu. Jigsaw: Supporting investigative analysis through interactive visualization. *Information Visualization*, 7:118–132, 2008.

[35] Voreen. Volume rendering engine for interactive visualization of volumetric data sets. http://www.voreen.org/.

[36] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.