G-PARE*: A Visual Analytic Tool for Comparative Analysis of Uncertain Graphs

Hossam Sharara†
Computer Science Department
University of Maryland, College Park

Awalin Sopan
Computer Science Department
University of Maryland, College Park

Galileo Namata
Computer Science Department
University of Maryland, College Park

Lise Getoor
Computer Science Department
University of Maryland, College Park

Lisa Singh
Computer Science Department
Georgetown University, Washington DC

ABSTRACT

There are a growing number of machine learning algorithms which operate on graphs. Example applications for these algorithms include predicting which customers will recommend products to their friends in a viral marketing campaign using a customer network, predicting the topics of publications in a citation network, or predicting the political affiliations of people in a social network. It is important for an analyst to have tools to help compare the output of these machine learning algorithms. In this work, we present G-PARE, a visual analytic tool for comparing two uncertain graphs, where each uncertain graph is produced by a machine learning algorithm which outputs probabilities over node labels. G-PARE provides several different views which allow users to obtain a global overview of the algorithms output, as well as focused views that show subsets of nodes of interest. By providing an adaptive exploration environment, G-PARE guides the users to places in the graph where two algorithms predictions agree and places where they disagree. This enables the user to follow cascades of misclassifications by comparing the algorithms outcome with the ground truth. After describing the features of G-PARE, we illustrate its utility through several use cases based on networks from different domains.

Keywords: Uncertain Graphs, Comparative Analysis, Model Comparison, Visualizing Uncertainty

1 INTRODUCTION

In today's linked world, graphs can be used to represent communication networks, social networks, financial transaction networks, gene regulatory networks, disease transmission networks, sensor networks and more. Different machine learning algorithms can be applied to observational data describing these networks to obtain predictive models. However, since the output is typically probabilistic and the input data is often noisy, it is not always clear which learning algorithm leads to models with the highest predictive accuracy or how different models compare. In this paper, we present G-PARE, an interactive tool for comparing the commonalities and differences among two predictive models and ground truth.

Graph analysis and visual display have a long and rich history; there are many visual analytics systems that have been proposed to support decision making over graph data. In this work, we look at a special type of graph, which we refer to as an *uncertain* graph, and the analytic task that we focus on is *comparative* analysis.

Our work is motivated by the desire to be able to understand the output of machine learning algorithms which operate on graphs. The generic category of algorithms we consider are *node labeling algorithms*, which take as input a partially observed graph, and output a graph with probability distributions over the unobserved labels of the nodes in the graph. We call this output graph an uncertain graph, because it has probability distributions over the node labels.

There are a wide variety of node labeling algorithms that have been proposed [9, 26, 30, 36]. Some machine learning algorithms may look only at attributes of the nodes to make predictions for labels [9], while others look at both the attributes and the labels of the neighbors [30, 36]. There are also simple algorithms which assign labels based on the proportion of neighbors with a given label [26]. Many of the models for disease spread and viral marketing can be cast as node labeling algorithms as well [24, 35].

This work supports the comparison of uncertain graphs that are output by different node labeling algorithms (which we refer to as models, for simplicity). This serves two main purposes: a) it helps the algorithm designer understand the dynamics of their node labeling algorithms and b) it can shed important light on a graph dataset, by showing where there is agreement among different models (this could be interpreted as areas where there is higher confidence in the predictions), and where there is disagreement, indicating a need to examine the data more closely, to see if there are errors or other interesting causes.

We have developed a visual analytic tool called G-PARE which supports the comparison of uncertain graphs. G-PARE uses a collection of views that allow users to see and compare the models output by different algorithms. It supports a variety of ways for finding where the models agree and where they disagree. It introduces a novel overlaid node-link diagram which supports showing both models and a comparison of their distributions on a single graph. It allows users to see the local neighborhood of a node, and quickly see if there is one node in isolation that has been misclassified, or if a large number of its neighbors have also been misclassified. The tool also allows users to follow "chains" of misclassification in the graph. In addition, the tool allows users to find larger regions in graph that are mostly agreeing or disagreeing.

The contributions of this work include: 1) an interactive visual analytic framework for uncertain graphs, 2) a comparative framework which supports macro-level comparative analysis for the entire graph, micro-level comparison of individual nodes, and meso-level comparison of neighborhoods around nodes, 3) a novel node visualization that captures uncertainty and comparison for the nodes in a node-link diagram, 4) a technique for allowing users to follow chains and cascades of misclassification, and 5) the description of several different use cases.

1.1 Uncertain Graph Data Model

The uncertain graph model that we introduce here is meant to be generic and simple, yet generic enough to capture a range of probabilistic semantics. As such, we are agnostic as to the exact underlying probabilistic model which produced the output. G-PARE uses

*website: http://www.cs.umd.edu/linqs/gpare
†e-mail: hossam@cs.umd.edu

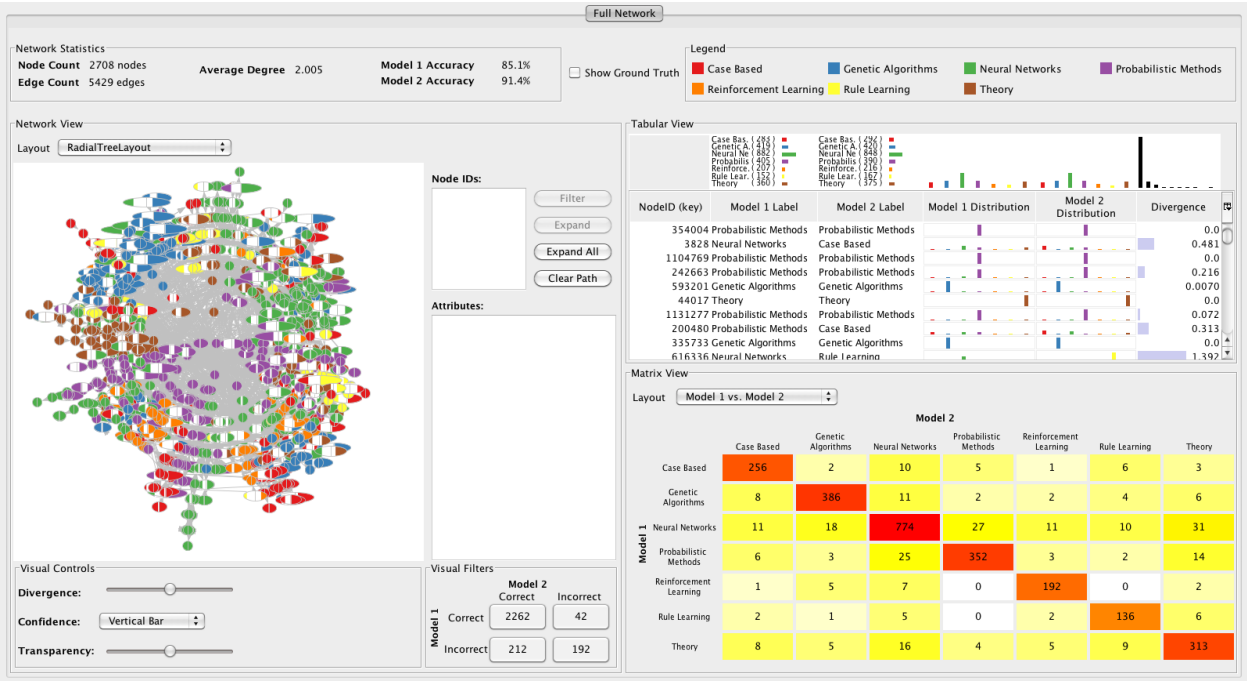


Figure 1: An overview of G-PARE applied to citation network data.

categorical labels, where the visualization is optimized for a range of labels from a few to a dozen or so; these are common ranges for the output of machine learning classification algorithms. Further, the label probabilities do not have to be the output of a machine learning algorithm; they can represent the assessment of a domain expert, or can be extracted from data in some other way. We do, however, for the purposes of this paper, assume that the only uncertainty is over the node labels (this is sometimes called *attribute uncertainty*) and the edges do not have uncertainty associated with them.

2 Related Work

There is a long tradition of work on uncertainty including defining the sources, types, and general approaches in visualizing uncertainty [15, 32, 34]. Pang et al. [32] define three general sources of uncertainty: data acquisition (through imprecise measurements, numerical models, or data entry), data transformation (through resampling and computers with limited precision), and visualization (due to errors and limitations in rendering the visualization). G-PARE addresses all three sources of uncertainty, providing means of comparing data acquisition and transformation uncertainty output, as well as providing multiple ways of visualizing uncertainty (through coordinated views) to minimize uncertainty through visualization. Pang et al. also define three categories of uncertainty: statistical (based on probability and confidence), error (based on differences between estimates and actual values), and range (intervals of possible values). G-PARE handles statistical and error uncertainty by supporting not only comparisons among probabilistic outputs, but also comparison to ground truth. Exploring ways to address range uncertainty is part of future work. Pham et al. [34] summarizes techniques for visualizing uncertainty including intrinsic representations (e.g., texture, color, size, shape), related representations (e.g., boundary, transparency), and extrinsic representations (e.g., bar graphs, histograms). G-PARE provides a way of visualizing uncertainty differences using a novel combination of multiple intrinsic and relation representations on the network view. G-PARE also applies extrinsic representations (shown in the tabular and matrix views) through the use of coordinated views.

Likewise, there is long tradition of work on visualizing networks. A number of interactive visualizations for networks have been proposed [1, 4, 7, 17, 29, 33, 37], along with different toolkits [3, 18, 31]. Most approaches use a node-link diagram as the basis for their visualization. Exceptions include ManyNets [12] which uses a tabular interface to analyze different node, edge and network statistics, and NodeTrix [19] which integrates matrices of the network connectivity and a node-link diagram. Our work builds upon these different traditions; our network view uses prefuse toolkit and our tabular view is inspired and partially developed on ManyNets. Similar to D-Dupe [6], C-Group [20], SocialAction [33], and NetClinic [25], G-Pare integrates data analysis techniques with graph visualizations.

A more recent trend is the interest in using visual analytic tools to better understand the strengths and weaknesses of models generated by different machine learning algorithms. WEKA [16], a suite of machine learning software developed by the University of Waikato, incorporates a set of illustrative plots and charts for visualizing the output of different machine learning models. iVisClassifier [10] makes the output of the classifier interpretable by presenting the pairwise cluster distances in a heatmap matrix, showing the attribute values of different dimensions in parallel coordinates after dimensionality reduction. Orange [11] presents a framework for visual exploration of the data and the model, providing various visualization options such as scatterplots, dendograms, trees, etc. Migut and Worring [28] also present an approach to visualize both model and data along with a framework to interact with the system in order to improve the model. ManiMatrix [21] provides a matrix interface to adjust the costs associated with different types of misclassification and lets the users refine the model accordingly. Similar to our approach, EnsembleMatrix [38] also uses a heatmap-like confusion matrix. It provides visual comparison method of the output of several models with the ground truth and lets the users adjust the weight of each model to create an ensemble learner.

While all these tools support analysis of predictive models, none of them support analysis of uncertain graphs. The closest work to

doing this is Cesario et al. [8] which proposes a set of linked views (parallel coordinates, bullseye, etc.) that highlight uncertainty of node labels in a single uncertain graph. While useful for seeing the distribution of node attribute labels, detailed analysis is difficult since the views support only aggregate comparisons, and there is no interactive component for selection and filtering.

3 Tool Description

G-Pare provides an interactive environment for the users to explore the commonalities and differences between two uncertain graphs. G-Pare also supports comparing the models to the ground truth when it is available. G-Pare is composed of three coordinated views: a tabular view, a matrix view, and a network view. The views provide different levels of detail, allowing users to understand differences at the aggregate model level, the uncertain graph neighborhood level, and the detailed node level.

G-Pare is written in Java™ and uses various toolkits to support building the different views. The network view is based on the prefuse visualization toolkit [18]. The tabular view, which shows the predicted labels and the distributions of compared models, builds upon ManyNets [12]. Finally, the matrix view uses standard Java™ graphical components for implementing the confusion matrix.

The main system architecture of G-Pare is based on two core components: a data access API and a visualization manager. The data API is responsible for obtaining network data and model information from a user-specified data source. The visualization manager is responsible for handling the creation and management of the UI, as well as the coordination across the different views.

To illustrate the features of G-Pare, we will use a document citation dataset as a motivating example. In this dataset, there are documents, and the links between documents represent citation links. The documents can have one of seven possible topics ("Case Based", "Genetic Algorithms", "Neural Networks", "Probabilistic Methods", "Reinforcement Learning", "Rule Learning", "Theory"). We will compare two different algorithms for predicting the topic/label of a document. The first algorithm uses only the words in the document to predict the topic. The second algorithm uses the topic labels of the documents which cite or are cited by the document. We refer to the output of the first algorithm as *Model1* and the second as *Model2*.

Figure 1 shows G-Pare interface, visualizing the two models output using the described citation network. The network statistics panel in the upper left shows summary information about the underlying network, including the number of nodes, the number of edges, and the average degree. When ground truth is available, the overall accuracy of both models' predictions is displayed in the same panel. Finally, the legend panel is located in the upper right of Figure 1. Here, node label values are mapped to different colors. These colors are used consistently across both the network view and the tabular view, allowing for visual coordination across views.

3.1 Tabular View

The tabular view (Figure 2) provides a side-by-side comparison of the models' predictions at the node-level. Each row in the tabular view corresponds to a node in the underlying network. The columns show information about the nodes according to one of the models. The first column shows the identifier for each node in the network. The next two columns show the most probable label for the node according to *Model1* and *Model2*, respectively. In cases where the ground truth is available, there will be another column following these two that shows the true label for the corresponding node. The following two columns show the probability distributions over all possible node labels under the two models. This distribution is represented as a color-coded histogram, where the height of each bar represents the probability that a specific node label is the 'true' label

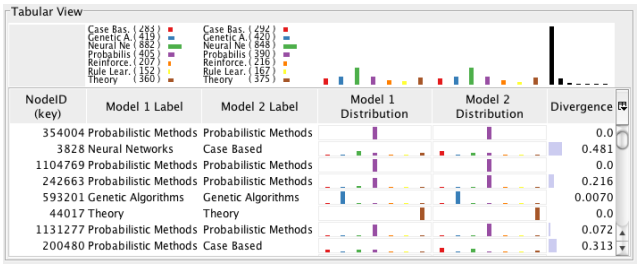


Figure 2: Tabular View

for the node. The last column shows a distance measured between the two distributions. We use symmetric Kullback-Leibler (KL) divergence [23] between the two probability distributions, but other measures are possible. Above each column header is a visual summary of the information contained in the column across all nodes. For example, the second column summarizes the number of times the model selects each node label. This allows users to quickly compare both the node details and the summary information within the same view. Lastly, the tabular view also supports common operations such as sorting, filtering, and multiple-selection of the nodes.

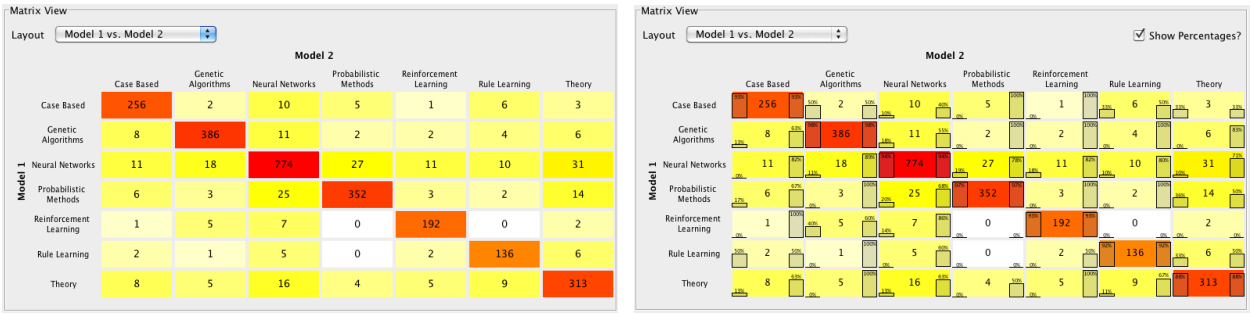
As an example of the kind of comparative analytics the tabular view supports, consider the first two rows in Figure 2, corresponding to publication numbers 354004 and 3828. For paper 354004, we quickly see that both models agree about the most probable node label; the two models agree that the most likely label is "Probabilistic Methods" (all of the probability mass is in that column) and the KL divergence between the distributions is 0. For paper 3828, we see that the two models disagree about the most probable node label: *Model1* predicts the paper to be a "Neural Networks" paper, while *Model2* predicts "Case Based". Furthermore, for this example, there is a considerable difference between the probability distributions of the node labels inferred by the two models. Further investigation of the probability distributions show that both models' predictions have low confidences. This can be seen in both the distribution column summaries and by noticing that there is a high KL-divergence score between the two models.

3.2 Matrix view

While the focus of the tabular view is detailed local information about nodes, the matrix view provides a more global, aggregate view of the models, highlighting areas where the two models agree and disagree. The matrix view shows a confusion matrix for the models' predictions. As shown in Figure 3(a), each cell (i, j) in the matrix shows the number of nodes whose predicted label is L_i according to *Model1*, and L_j according to *Model2*. The main diagonal in the matrix corresponds to cases where the two models agree on their predictions, while the off-diagonal cells are the places where they disagree. We use a heat map visualization of the counts to further highlight the cell frequencies. The color of each cell ranges from white to red, logarithmically according to the cell count, with red indicating the highest frequency and white corresponding to the lowest one. This view lets users quickly see the number of nodes where the two models agree and disagree.

In addition to providing an overview of the commonalities and differences between the predictions of the compared models, the matrix view also supports selection and filtering. It allows users to zoom into areas of interest by selecting a given cell of the matrix, and inspecting the characteristics of the selected nodes further in filtered tabular and network views.

For our citation network example, Figure 3(a) shows that the majority of publications falls along the main diagonal, indicating that, for the most part, the two models agree on their predictions. For in-



(a) The matrix view for the citation network

(b) Overlaying the ground truth on the matrix view

Figure 3: Matrix View

stance, the"Neural Network" topic has the largest number of matching predictions between the compared models, with a cell count of 774. Looking at places where the models disagree, we see that there are 31 publications that are labeled "Neural Network" by *Model1*, and "Theory" by *Model2*.

In cases where the ground truth is available for the underlying data, the matrix view provides users with additional options to compare the models to the ground truth. In addition, the matrix view shows the models' accuracy for a particular label. A small histogram is introduced to each cell in the matrix, as shown in Figure 3(b), with two vertical bars corresponding to the percentage of correct classification for each model. For example, cell(Probabilistic Methods, Neural Networks) shows that *Model1*'s accuracy is 20%, while *Model2*'s accuracy is 68%. This can be interpreted as, among the 25 papers that *Model1* predicted to be "Probabilistic Methods" and *Model2* predicted to be "Neural Networks", 5 papers are actually "Probabilistic Methods", 17 are "Neural Network" papers, and 3 papers were misclassified by both models and have neither topic. More subtly, users not only know that 17 papers were accurately classified by *Model2*, but also that *Model1* misclassified them as having the topic "Probabilistic Methods". Sometimes understanding how a label is being predicted inaccurately is important for understanding the weaknesses of a model.

3.3 Network View

While the previous two views provide useful micro (node-level) and macro (aggregate-level) tools for comparing two models, an important aspect that is missing is the ability to view and compare the networks directly, especially the neighborhoods around nodes or a collection of nodes of interest. G-PARE's third view, the network view, provides this capability. It shows the node-link diagram of the underlying graph, and supports multiple types of network layouts, such as the radial tree layout, the force-directed layout and Fruchterman Reingold layout [13].

The novelty of the network view is the overlay of information from the two models into a single view. Each node in the network is represented as an ellipse, where different visual properties are used to encode various features such as the most probable label, the probability of the label, the divergence of the two models, and the comparison with ground truth. As shown in Figure 4, every node is split in half, with the left side corresponding to *Model1* and the right side corresponding to *Model2*. The color of each node half corresponds to the most probable label of the corresponding model. This allows the user to quickly identify nodes where the two models agree in their predicted label (same color), and the ones where they disagree (two-tone). The ellipse eccentricity is used to encode the divergence between the probability distributions of the two models. The ratio by which the ellipse eccentricity correlates with the KL-divergence is user-controllable through a slider in the visual controls panel, as shown in Figure 5(b). G-PARE provides users with multiple options to visualize the model's confidence in the predicted label. Figure 5(a) shows the different options users has for encoding the confidence, including using a vertical bar, using a horizontal bar, and changing the intensity of the filled areas.

Finally, if ground truth is available, users can use the thickness of the node borders to highlight the half-ellipses that correspond with a model making a correct prediction. This helps users quickly identify which nodes are correctly predicted by both models (solid border highlight), which are correctly predicted by one (a two-tone node with a half ellipse highlighted), and which are incorrectly predicted by both (an unhighlighted border, either solid or two-tone).

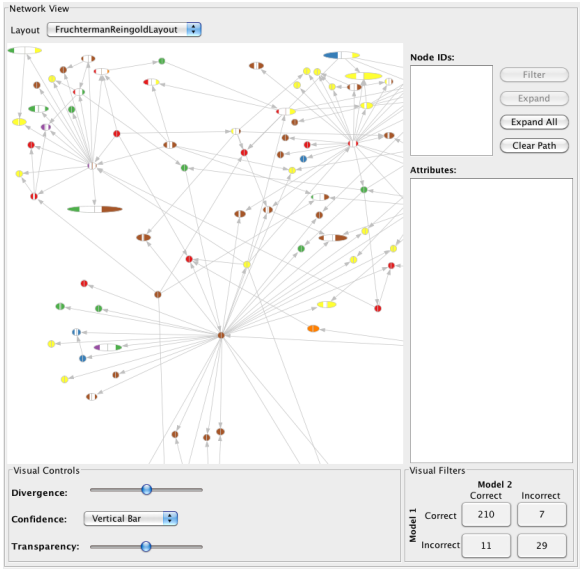
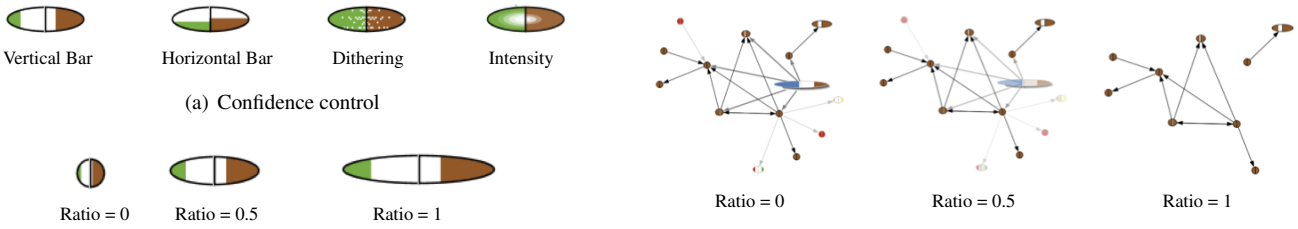


Figure 4: The citation network shown in the network view.

In cases where the ground truth is known, the network view of G-PARE provides users with additional capabilities for selecting nodes based on the accuracy of the compared models. As shown in the bottom of Figure 4, there is a 2x2 matrix that shows the counts of nodes where both models' predictions are correct, places where both are incorrect, and places where one is correct and the other is incorrect. The cells in the matrix are clickable, and will select the corresponding nodes.

When nodes are selected, either through node selection on the graph or the above mentioned mechanism, the network view also includes an information panel that is used to display the attributes of any selected set of nodes. In addition, when a set of nodes is



(a) Confidence control

(b) Divergence ratio

(c) Transparency ratio

Figure 5: Visual controls used for encoding confidence, divergence, and focus

selected, the rest of the network is dimmed to bring the focus to the selected nodes. This helps users focus the analysis on specific parts of the network, without losing the information about the position and the connections of the target set with respect to the global network. Similarly, users can control the percentage of dimming through a dedicated visual control as illustrated in Figure 5(c).

In addition to the various filtering and highlighting options that help users focus on areas of interest in the network, G-PARE also utilizes the zoom and translation features provided by the underlying prefuse toolkit [18] to provide users with means for exploring the network without resorting to filtering. These features also help users overcome the cluttering that can occur in visualizing dense networks, by navigating through different parts of the graph. Paired with the different network layouts supported by G-PARE, these exploration features allow users to gain different insights about the comparison of the models in different regions of the graph.

Figure 6 illustrates an example from the citation network. Using the visual encoding of the selected node, we can infer the following:

- *Model1*, corresponding to the left half-ellipse, predicts the paper topic to be "Neural Networks" with a low probability.
- *Model2* predicts the paper topic to be "Theory" with higher probability.
- The divergence between the probability distributions of the two models is high.
- The border-highlighting shows that *Model2* is making the correct prediction.

In addition, the selected paper's attributes (true label and word occurrence) are listed as well.

3.4 Interaction

As is often the case with network visualization, viewing the entire network helps provide an overview of the underlying data, but it is not very informative for analytical tasks. At the global level, G-PARE helps users navigate through the underlying data and models to identify areas of interest by providing different levels of abstraction in each of the three described views. The full coordination among the tabular, matrix, and network views allows the user's selection from any view to be applied as a filter on the remaining two. This allows users to make a selection in one view (e.g., an entry in the confusion matrix) and observe the selected set of nodes in another view, which gives a more detailed view (e.g., the connection pattern between the nodes corresponding to the confusion matrix entry).

After the user has selected a set of nodes of interest using any of the three views, the user can then filter these nodes out for further analysis. By clicking on the filter button, a new window, with a new set of views, is created over the selected nodes. Both the original

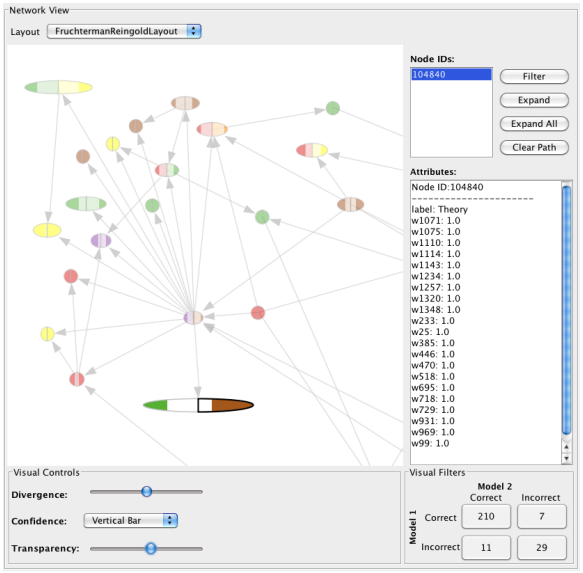


Figure 6: Illustrative example showing a selected node, its predicted label by the compared models, and the neighborhood around it.

selection and the new window show as different tabbed windows in G-PARE. In the node-link diagram in the new window, the nodes that were used to construct the view are shown with a shadow, so that users can keep track of the original selections, as further exploration is performed.

The network view provides additional functionality for expanding the nodes shown in the node-link diagram. By clicking on the "Expand" button in the network view, all the neighboring nodes of the selected nodes (e.g., their ego-networks) are added to the current set. On the other hand, clicking the "Expand All" button adds the neighbors of all the nodes currently in the network, irrespective of them being selected, to the current view. This process can be repeated, so, with enough expansions, starting from a single node, users will be able to get the complete connected component in which a node participates. More often in analysis, users will want just the "1-hop" or "2-hop" neighbors of the nodes of interest.

As an example of filtering, in many cases the nodes of interest are ones with the highest KL-divergence between the models' distributions. By sorting the rows of the tabular view by divergence, the top nodes can then be selected, filtered, and shown in a new window. Figure 7 shows the expanded ego networks of the two nodes in the running example with the highest KL-divergence. By overlaying the ground truth, we can observe the following: 1) For the right node, we see that *Model1*'s prediction, based on paper content,

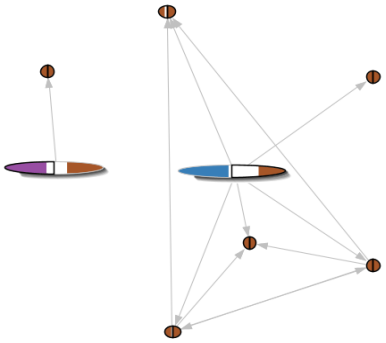


Figure 7: A filtered view of the two nodes with the highest divergence.

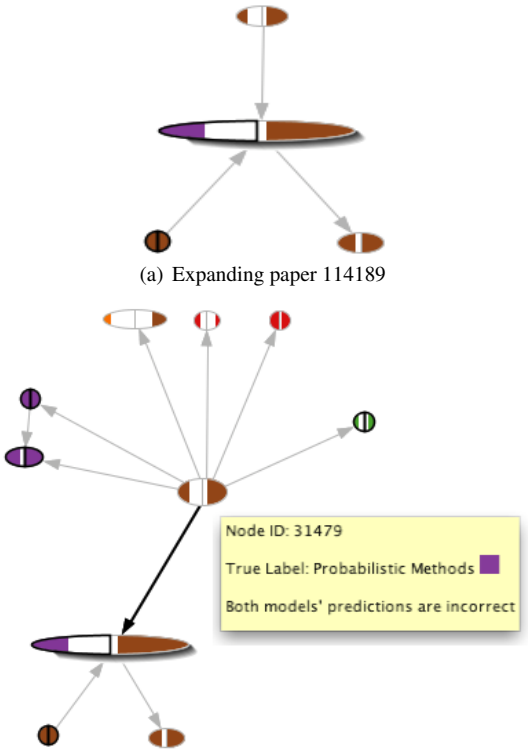
indicates that the paper's topic is "Genetic Algorithms." However, *Model2*, which takes relational structure into account, shows that most of the citations to and from the paper are to "Theory" papers. Based on this, *Model2* predicts the paper's topic to be "Theory," which is the correct topic, as shown by the border highlighting of *Model2*'s prediction. 2) For the left node, the same signal from the node's neighbors causes *Model2* to misclassify the paper's topic as "Theory." In this case, the true topic is "Probabilistic Methods," as predicted by *Model1*. Intuitively, one can think that the signal the left node is receiving from one neighbor should not be as strong as the signal the right node is receiving from 5 neighbors. In fact, *Model2* uses the percentage of neighbors with a given label, rather than the actual count; a new model, which uses count might perform better. Thus, G-PARE is able to reveal some shortcomings in the underlying models, which can then be taken into account in interpreting the results or refining the model.

Another feature that G-PARE provides is the ability to follow a given path through the network. This allows users to identify patterns, detect cascades of errors, etc. Figure 8 shows an example of utilizing the path-following feature in the bibliographic dataset. By investigating the ego-network of paper 114189 in Figure 8(a), we can see that the paper is misclassified as "Theory" by the relational model since all the papers that it is connected to are "Theory" papers. However, by overlaying the ground truth, two of the three neighbors of the paper are actually misclassified as well. In order to identify the error source, we can then expand the top neighboring node to also include its ego-network. As shown in Figure 8(b), G-PARE keeps track of the current path that the user is exploring by highlighting the corresponding nodes and edges. By examining the neighborhood of the newly expanded paper (31479), we observe that 50% of its neighbors are misclassified as "Theory" papers as well by the relational classifier. By hovering over paper 31479, we can see that the true topic of the paper is "Probabilistic Models", similar to the true topic of paper 114189.

One thing we note here is that paper 31479 has 33% of its neighbors correctly classified as "Probabilistic Models" papers by both models, with higher probability than the misclassified neighbors. Thus, we again hypothesize that if *Model2* were to take the output probability of the neighbors' predictions into account, it would have made a better prediction in that case. We can continue following the error cascade by expanding the neighborhood of paper 31479 further, as shown in Figure 8(c), until we reach the source of the cascade to investigate the source of the error.

4 USE CASES

This section presents three use cases, highlighting different uses of the tool.



(a) Expanding paper 114189

(b) Expanding paper 31479

(c) Expanding paper 14549

Figure 8: Following a cascade of errors

4.1 Communication network use case

In this use case, we apply G-PARE to the internal email communications of a former major energy company, the Enron Corporation. The email communications of the Enron Corporation were released in 2003 as part of a Federal Energy Regulatory Commission (FERC) investigation into Enron accounting practices [22]. As one of a few publicly available email communication datasets, the dataset provides a unique glimpse into the interactions of different types of individuals in a major organization. An important aspect of these interactions is how indicative the content (i.e., their word

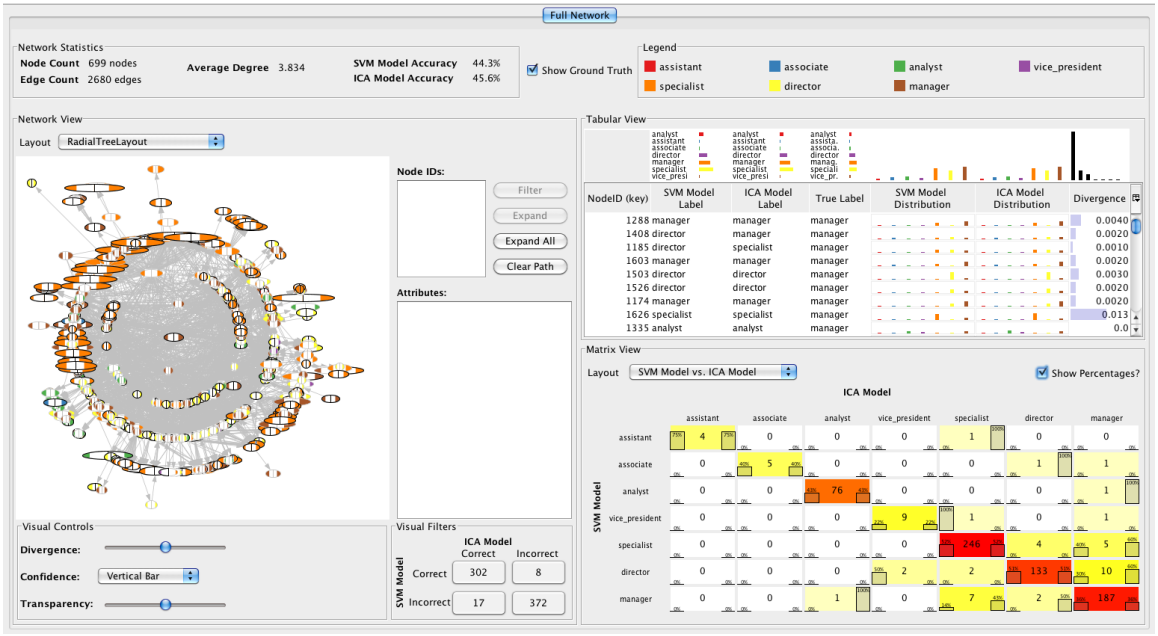


Figure 9: Initial display for the communication network case study. The nodes represent email addresses, label indicates the title of the email address user, and edges indicate communication exchanged between email addresses.

usage) and relationships (i.e., with whom they share emails) are of an individual's position (i.e., title) within the organization.

In this case study, we use a subset of the USC version of the dataset [2] along with information from an internal Enron document which lists the titles of individuals in the Enron Wholesale Group. We merge multiple levels of the same general position (e.g., junior specialist to specialist) and exclude titles which have less than 100 instances. In the network used in this study, nodes represent the email addresses with known titles. Directed communication edges are added between email addresses which have shared at least 5 communications. We also remove nodes which do not have at least one communication edge. Next, word features are created from the email communications sent between these email addresses which (after stemming, stop word removal, and filtering based on frequency). The final network consists of $1,402$ email address nodes with $9,523$ communication edges, 7 possible titles (assistant, associate, analyst, vice president, specialist, director, and manager), and 500 binary word indicators over the nodes.

Our use case compares two models for predicting the titles of the participants. The first uses a state of the art classifier, support vector machines (SVM) [9], based on word usage. The second model is a collective classification model which not only uses the content of the emails, but also the titles of the individuals with whom they share communications. We compare these models by splitting the full network into two disjoint splits using snowball sampling, training both models on one split, and applying the learned models and comparing the output on the other.

The initial display of G-PARE is shown in Figure 9. We begin by first looking at the overview provided by G-PARE to describe the network to which we applied our models. We can see that this network consists of 699 nodes and 2680 edges, with an average degree of 3.8. For the output of the models, we see that SVM has an accuracy of 44.3% and the collective classification algorithm has an accuracy of 45.6%. While this shows that both models improve over a random baseline (which would have an accuracy of 14.3%), it also shows there is room for improvement and a need to understand why the incorrect predictions are occurring.

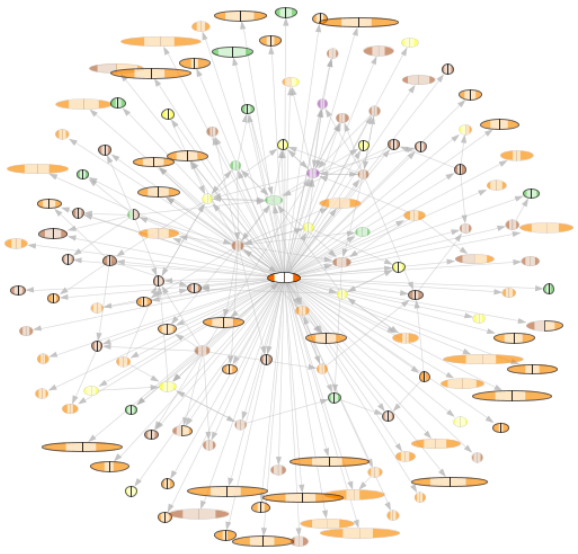


Figure 10: Ego network of a highly connected individual in the network, 'Robert Ambrocik', whose predicted labels seems to impact many of the cases about which the models most disagree.

One possible cause can be seen when comparing the class distributions found in the tabular view. We see that while the predicted distributions of both models are generally consistent with the true class distribution, both models do tend to incorrectly predict the majority title, Specialists, more often than the number present in the ground truth. We see this bias further in the matrix view where the largest entry, 246 nodes, is for the set of nodes both models predicted as Specialist, 48% of which are incorrect. This observation

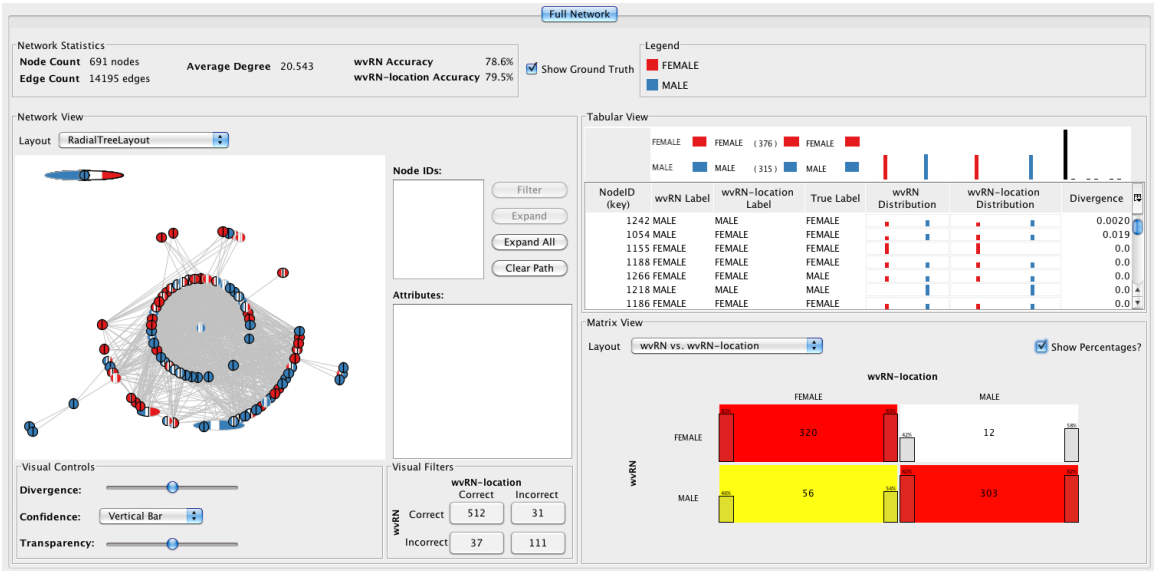


Figure 11: An overview of G-PARE applied to the dolphin data.

implies that skew in the number of instances for each title is likely to be affecting the quality of the models.

Focusing on where the models disagree, the matrix view shows that most disagreements are about the titles of Director, Manager, and Specialist. These disagreements imply that the characteristics of people with these roles, in terms of word usage and relationships, are likely very similar. Viewing the comparison of both models to ground truth, we see a similar difficulty distinguishing between Analysts and Associates. This implies that the features we are using may be insufficient for our prediction task and that more complex models (e.g., using bigrams, more complex notions of structural equivalence) may be needed.

Aside from guiding us to possible ways to improve the models, the cases where both models disagree also highlight anomalous individuals whose interactions may be of interest from a sociological standpoint. For example, there is an individual, 'Jerry Britain', whose word usage seems to indicate he is an Assistant while his relationships correctly indicate he is a Specialist. The same can be said of the 'James Gilbert', whose word usage indicates he is an Associate while his relationships correctly shows he is a Director. Comparing their word usage to others whose word usage correctly indicates their position may provide insight into language differences across positions.

Finally, by sorting by the Divergence column, we notice an interesting trend among the cases where the probability distributions of the two models diverge. In the 12 most divergent cases, when we view their ego networks, we see that all these nodes only have a degree of one. More interestingly, of these, 10 share their edge with the same individual, 'Robert Ambrocik.' Filtering and viewing his ego network, shown in Figure 10, we see that 'Robert Ambrocik' is a well connected individual in this network whose label has had impact on many of the cases where the two models most disagree (highlighted by the eccentricity of related nodes). This observation not only provides further insight into the sensitivity of our collective model to node degree, but also shows the impact a prediction on a single node can have on neighboring nodes.

4.2 Bottlenose dolphin use case

In this use case, we consider a network based on a long-term study of a wild bottlenose dolphin population in Shark Bay, Australia [27]. Scientists that study this bottlenose dolphin community have found that behavioral differences exist across the sexes [14]. Therefore, knowing the sex of the dolphin can be important for understanding different behaviors within the population. The sex of the dolphin is determined by observation when possible, e.g., when a dolphin leaps or swims on its back, and DNA sampling in some cases. Because the number of dolphins monitored in this population is large (over 1200), and some dolphins are not seen often, the sex of 40% of the dolphins is unknown. Therefore, it is useful to investigate methods for predicting the sex of dolphins given previous observational data about the dolphin population. To support this task we use observed dolphin associations to build predictive models for inferring the sex of a dolphin. We focus on observational surveys collected by the researchers.

In the dolphin network, dolphins are represented as nodes, and edges exist between dolphins that have been observed together. In our problem settings, the node labels represent the sex of the corresponding dolphin. We begin our analysis by comparing two models, one that decides on a node label using the predicted labels of the node's neighbors (*Model1* - wvRN) and one that uses the predicted labels of the neighbors having the same predominant observed location (*Model2* - wvRN-location). Figure 11 shows the initial display in G-PARE. We see that the accuracy of the two models is very similar, 79% and 80%, respectively. The matrix view shows us that the models agree on the labels for 320 female and 303 male dolphins.

Using different filters, we can focus on the subset of nodes that are mislabeled by both models. By further examination, we note that the average degree among the filtered subset of nodes is 3.5, which is significantly lower than the average degree of the full network 20.5. This may indicate that there exist fewer observations for these dolphins than other ones.

Given this new information, we generate new models (Model 3 - wvRN) and (Model 4 - wvRN-location) that train on data with dolphins with at least 5 observations. We see that the accuracy of both models improves to 83% and 82.8% respectively. We find that *Model3* misclassifies males and females at about the same rate, while *Model4* misclassifies females at a higher rate than males, indicating that predominant location does impact the mislabeled nodes (even though the overall accuracy is about the same). On further investigation, many of the females that are being misclassified by *Model4* swim in different locations with different dolphins. Therefore, using the predominant location does not help the prediction

accuracy for these dolphins. Without exploring the data, analysts may incorrectly assume that location had no impact on the final model. The ability to see the mislabeled nodes and their neighborhoods gives the analyst insight into where different models perform poorly and where they perform well.

4.3 Citation network use case

This case study considers a second citation dataset, gathered from Citeseer, a search engine and digital library for scientific and academic papers. For our use case, we use a portion of the full Citeseer network which contains 2,120 nodes, 3,757 edges, a 3703 word vocabulary, and a label indicating the topic of a paper (i.e., AI, Agents, DB, HCI, IR, ML).

In this use case, we look at the outputs of two common approaches for this document classification. The first (*Model1* - SVM) uses only the document content and using a support vector machines (SVM) predicts the paper topic. The second (*Model2* - Majority) uses the topic labels of the neighbors for prediction. In our case study, we use a simple algorithm which repeatedly iterates over the nodes of the network, labeling each unlabeled node with the most common label among papers which it cites or is cited by (using labels observed or predicted in previous iterations), until all the nodes are labeled.

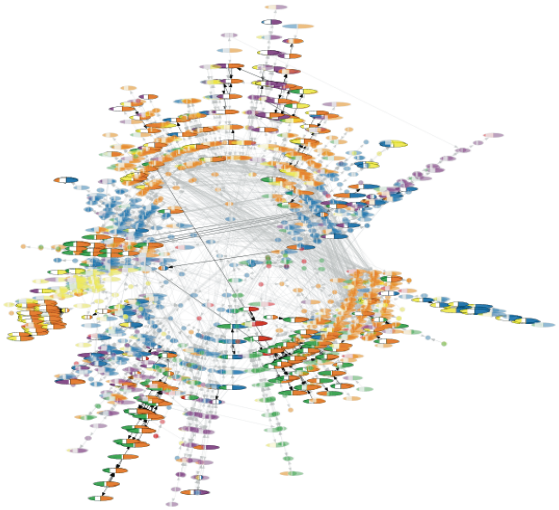


Figure 12: Network view showing an overview of the Citeseer citation network where nodes predicted by *Model1* correctly and by *Model2* incorrectly are highlighted. The overview shows large areas of misclassification to the same label, a phenomenon referred to as *flooding*.

After loading the dataset, we first notice that both models perform well, with accuracies of 75% and 67% respectively. More striking, however, is that while both models perform well, when looking at the overview of class distributions and divergence in the tabular view, we see that the predictions of both models have very different characteristics. Selecting the option to display the ground truth, we see that the overall class distribution of Model 1 is fairly consistent with the ground truth but that Model 2 is heavily skewed toward two topics, Agents and IR. We see the same in the network view, show in Figure 12, with most of the network colored blue and orange, corresponding to Agents and IR respectively. Of note, however, is that the labels are not distributed evenly throughout the network. Instead, we see large interconnected portions of the network which are all predicted by *Model2* as the same label. We see this phenomenon further when we filter on the cases where *Model1* is correct but *Model2* is incorrect. As show in Figure 13, we find a

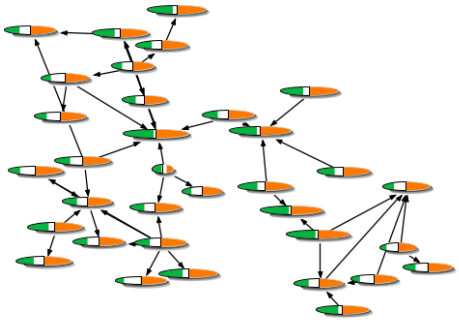


Figure 13: A subnetwork of the Citeseer citation network which illustrates *flooding*. While *Model1* correctly predicts the node labels as DB (shown in green), *Model2* incorrectly predicts them as IR (shown in orange).

connected component on 34 nodes which *Model2* incorrectly predicts as IR (show in orange) and which *Model1* correctly predicts as DB (shown in green). This result is consistent with a known problem in relational models called *flooding* where incorrect neighboring labels can cause cascades of errors [5]. By using G-PARE, we are not only able to find a real world example of this little studied phenomenon, but we can also expand out from these cascades to uncover the initial source of the misclassifications.

5 CONCLUSION AND FUTURE WORK

We have described G-PARE, a visual analytic system designed to support users in comparing uncertain graphs. The visual components of G-PARE combine elements from network and uncertainty visualization, the most novel of which is the node visualization which captures both comparison and uncertainty information. The interactive components of G-PARE highlight the abilities to examine and compare models at the macroscopic (full graph), microscopic (node in the graph) and mesoscopic (neighborhood around a collection of nodes) level. While G-PARE was designed with the goal of comparing the output of machine learning algorithms, there are other settings (such as the comparison of the probabilistic evaluations of experts) where comparing uncertain graphs makes sense. There are a number of areas open for future research. One is supporting richer uncertain graph models, such as uncertainty over edges, as well as supporting unaligned graphs – here we assumed that both uncertain graphs were over the same node set, so that the node mapping was given. Additionally, we plan on investigating coupling the tool with the underlying models, where the user is allowed to provide feedback that is incorporated for enhancing the models in real-time. Nonetheless, in this work, we believe we have developed a unique tool that is well-suited to the analytic task of understanding differences and commonalities between node labeling algorithms.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their valuable feedback, Catherine Plaisant and Ben Shneiderman for helpful comments during the development of the system, Janet Mann and the Shark Bay Dolphin Research Project (SBDRP), and our FODAVA partners. This work was supported in part by FODAVA grants from NSF Grant #CCF0937094 and Grant #CCF0937070.

REFERENCES

[1] E. Adar. Guess: a language and interface for graph exploration. In *International Conference on Human Factors in Computing Systems*, pages 791–800, 2006.

[2] J. Adibi. Enron dataset. Dataset available at http://www.isi.edu/ adibi/Enron/Enron.htm.

[3] 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.

[4] S. Bender-DeMoll and D. A. McFarland. The art and science of dynamic network visualization. *Journal of Social Structure*, 7(2):1–38, 2006.

[5] M. Bilgic and L. Getoor. Reflect and Correct: A misclassification prediction approach to active inference. *ACM Transactions and Knowledge Discovery from Data (TKDD)*, 3(4):1–32, 2009.

[6] 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.

[7] P. Caravelli, M. Beard, B. Gopalan, L. Singh, and Z.-Z. Hu. Generating abstract networks using multi-relational biological data. In *International Conference on Information Visualisation*, pages 331–336, 2009.

[8] N. Cesario, A. Pang, and L. Singh. Visualizing node attribute uncertainty in graphs. In *Society of Photo-Optical Instrumentation Engineers Conference Series*, 2011.

[9] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[10] J. Choo, H. Lee, J. Kihm, and H. Park. iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 27–34, 2010.

[11] 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*, volume 3202, pages 537–539, 2004.

[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*, pages 213–222, 2010.

[13] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software-Practice and Experience*, 2:1129–1164, 1991.

[14] Q. Gibson and J. Mann. Early social development in wild bottlenose dolphins: Sex differences, individual variation and maternal influence. *Animal Behaviour*, 76:375–387, 2008.

[15] H. Griethe and H. Schumann. The visualization of uncertain data: Methods and problems. In *Conference on Simulation and Visualization*, pages 143–156, 2006.

[16] 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.

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

[18] 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*, pages 421–430, 2005.

[19] 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.

[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*, pages 211–212, 2007.

[21] A. Kapoor, B. Lee, D. S. Tan, and E. Horvitz. Interactive optimization for steering machine classification. In *International Conference on Human Factors in Computing Systems*, pages 1343–1352, 2010.

[22] B. Klimt and Y. Yang. Introducing the enron corpus. In *Conference on Email and Anti-Spam*, 2004.

[23] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[24] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. In *ACM Conference on Electronic commerce*, 2006.

[25] 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*, pages 131–138, 2010.

[26] S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research*, 8:935–983, 2007.

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

[28] M. Migut and M. Worring. Visual exploration of classification models for risk assessment. In *IEEE Symposium on Visual Analytics Science and Technology*, 2010.

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

[30] J. Neville and D. Jensen. Iterative classification in relational data. In *AAAI Workshop on Learning Statistical Models from Relational Data*, 2000.

[31] 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.

[32] A. Pang, C. M. Wittenbrink, and S. K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370 – 390, 1997.

[33] A. Perer and B. Shneiderman. Balancing systematic and flexible exploration of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):693 – 700, 2006.

[34] B. Pham, A. Streit, and R. Brown. Visualisation of information uncertainty: Progress and challenges. In L. Jain, X. Wu, R. Liere, T. Adriaansen, and E. Zudilova-Seinstra, editors, *Trends in Interactive Visualization*, Advanced Information and Knowledge Processing. Springer London, 2009.

[35] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*, pages 61–70, 2002.

[36] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[37] 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.

[38] J. Talbot, B. Lee, A. Kapoor, and D. S. Tan. EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers. In *International Conference on Human Factors in Computing Systems*, pages 1283–1292, 2009.