Report on the First International Workshop on Mining Graphs and Complex Structures (MGCS'07)

Lawrence B. Holder
Electrical Engineering and Computer Science
Washington State University
holder@wsu.edu

Xifeng Yan
IBM T. J. Watson Research Center
xifengyan@us.ibm.com

1. INTRODUCTION

The fast accumulation of graph data is witnessed in a wide range of scientific and commercial domains. Typical graph data include chemical compounds, circuits, biological networks, computer networks, 2D/3D models, XML, RDF and workflows. Graph is regarded as a critical data type for knowledge discovery in bioinformatics, chemical informatics, computer vision, informational retrieval, computer security, semantic web, social science, etc., just to name a few. Unfortunately, due to the lack of graph management and mining tools, it is hard, if not impossible, for users to search and analyze any reasonably large collection of graphs. There is an imminent need for scalable methods for mining and search in graphs and other complex structures.

The First International Workshop on Mining Graphs and Complex Structures provides researchers a forum on the new development of knowledge discovery in graph and complex data. It was organized by the Seventh IEEE Int. Conf. of Data Mining (ICDM 2007) and held at Omaha, Nebraska. The workshop covers topics including, but not limited to, graph pattern mining, graph search, graph language, graph classification, link analysis, graph kernel method, social network analysis, etc. The workshop received 41 submissions and accepted 11 papers among them, which were presented in three themes: Clustering in Networks, Link Analysis and Classification, and Graph Pattern and Language.

2. WORKSHOP SESSIONS

2.1 Session I - Clustering in Networks

The ability to cluster documents into well-defined categories is an important task for organizing and understanding the vast number of documents available today. Most techniques addressing this task are based on an analysis of frequently co-occurring keywords within the documents. In "*GDClust: A Graph-Based Document Clustering Technique*", Hossain and Angryk have developed a new way of measuring the similarity of documents based on their sense, that is, their structural position within an ontology. This similarity is evaluated by generating a graph representation of each document, where edges in the graph represent a hypernym relationship if two words from the document reside in ontological sets with this relationship. Thus, the graph represents the structure within the ontology, which is independent of the specific keywords or their frequency. Results show that this approach produces a clustering of a real-world set of documents that closely resembles the known underlying categories of the documents. Such an approach, which relies less on the appearance of specific words, is more robust than traditional approaches and represents an advanced method for organizing the numerous documents available to us on a daily basis.

In addition to the task of categorizing a set of graphs, graph-based clustering also includes the task of clustering a single graph by identifying a partitioning of the vertices into sets with high inter-cluster distance and low intra-cluster distance. In "*A Divisive Hierarchical Structural Clustering Algorithm for Networks*", Yuruk et al. propose a distance measure based on the structural similarity of vertices, that is, two vertices are close if they share many neighboring vertices. They use this distance measure to evaluate each clustering resulting from an iterative removal of edges from the graph. The algorithm chooses the clustering that maximizes the ratio of the inter-cluster and intra-cluster distances, and therefore does not require any user parameters for guiding the choice for the best clustering. Results on three well-known datasets show that this approach finds clusterings that meet or exceed the quality of those found by alterative approaches. Such an algorithm eases our search for clusters within a graph and has application to many domains, including community identification in social networks to common functionality in biological networks.

Chen et al. propose a different approach to clustering networks by viewing the networks as a depiction of higher-order relationships in heterogeneous data. In "*Simultaneous Heterogeneous Data Clustering Based on Higher Order Relationships*", they propose a tensor model of the network, which is essentially a multi-dimensional matrix, where each dimension represents a different property used to describe objects, and the contents of the matrix defines a hypergraph among the vertices representing the objects. They then replace the hyperedges (edges connecting more than two vertices) with a clique defined over the vertices of the hyperedge and perform a more traditional edge-cutting approach to partitioning this graph. The result is a clustering of the objects that takes into account the higher-order relationships defined among the objects. They empirically verify the effectiveness of their approach, which has application to any dataset defined using objects with zero-order to higher-order relationships.

Once we have found clusters within or across networks, we would ideally like to describe these clusters in terms of the salient properties of the members of the cluster, or more

global properties of the entire cluster. Furthermore, the evolution of the cluster over time can also provide insight into the reason for the existence of a cluster. This qualitative description of the properties and evolution of a cluster is termed the resume of the cluster by Wu et al., in their work "*Resume Mining of Communities in Social Network*", and they present algorithms for extracting this information from clusters within a network. One of their main observations is that clusters are best identified by the stable characteristics of their core members over time rather than all members of the cluster. By identifying and tracking these core members, they are able to produce a resume for a network that helps explain the existence and present state of the network.

2.2 Session II - Link Analysis and Classification

While relational clustering seeks to categorize unclassified relational data, relational classification seeks to infer the class of unlabeled test data given some amount of training data. The relational classification task is complicated, as compared to non-relational classification, by the fact that instances may be related and therefore violate the independence assumptions underlying many non-relational learning approaches. In order to explore these relational classification issues, Gallagher and Eliassi-Rad view the problem as a network classification problem in their work "*An Examination of Experimental Methodology for Classifiers of Relational Data*", and divide the problem into two classes: between-network classification and within-network classification. Between-network classification involves learning a model from one relational network and then using this model to classify the nodes of another network. Within-network classification involves the training and testing nodes residing in the same network, possibly interconnected, and therefore classification of a testing node may draw upon the class labels of its neighbors. Classification may follow a similar learn-then-classify process as for between-network classification, or may use collective inference to iteratively refine the class labels based on the possibly changing class labels of neighboring nodes. The authors perform some empirical studies to understand the interdependencies of the different aspects of the within-network classification problem. One finding shows that the availability of labeled neighbors during the testing phase has a greater value than increasing the number of training examples. These results help us better understand the added complexities of evaluating relational classification methods.

One approach for improving the classification task is to remove some of the features that are deemed irrelevant or redundant. However, in some tasks (e.g., document classification) results show that feature selection has limited benefits. In "*Learning Term Dependency Links Using Information Theoretic Inclusion Measure*", Makrehchi and Kamel argue that the limited benefits to feature selection can be due to ignoring term dependency. They propose an information theoretic measure for determining the dependency among terms and then remove those features that are redundant given this dependency. Empirical results show that this approach outperforms the popular support-vector machine approach and a more aggressive feature selection scheme. In general, taking into account the relationships among features in a relational learning task can improve classification performance.

Collective classification is one method for addressing the within-network classification task; namely, the class of an unlabeled instance is based on its labeled neighbors. Collective classification continues iteratively until all nodes of the network are classified. The progress of the method can be viewed as the flow of information from the initially labeled nodes eventually to the unlabeled nodes. Given this flow view, we can consider the issue of which nodes, if labeled initially, have the most impact on the performance of collective classification. Since determining the correct label of nodes may be costly, we would like to select a small number of influential nodes. This selection of initially-labeled nodes is termed active inference, and Rattigan et al. ("*Exploiting network structure for active inference in collective classification*") consider alternative schemes and their relationship to the amount of autocorrelation (similarity in attributes of linked entities) present in the network. Of the schemes studied, the k-means approach of identifying locally-influential, yet globally-dispersed, nodes provides the best result. They also show that the influence of all schemes increases with the amount of autocorrelation. These results will help with the identification of nodes initially labelled in order to maximize the performance of collective classification.

In addition to the task of classifying nodes in a network, we may also need to predict the presence of a link in the network. Typically, collective classification of nodes and link prediction have been studied independently, but many real-world network classification tasks require both forms of inference. In "*Combining Collective Classification and Link Prediction*", Bilgic et al. explore the combination of collective classification and link prediction to see if their iterative application can improve both tasks. Using a synthetic data generator they were able to generate networks with varying amounts of autocorrelation, attribute noise, link noise and link density. Results show that the combination of collective classification and link prediction outperformed either method employed individually, suggesting that these methods should always be employed together. In this session, Bilgic, Gallagher and Jensen also exchanged their opinions on the challenging issues of link prediction that arise from high false positive error rate.

2.3 Session III - Graph Pattern and Language

Kernel-based learning methods have become some of the most successful learning methods for a variety of problems. Kernel methods work by transforming the feature space of the learning problem into a higher-dimensional feature space, where typically learning is easier. Planar languages represent a class of languages for which kernels exist that map strings into a point in the higher-dimensional space, and learning with planar languages has been shown to converge with only positive examples. However, strings are insufficient to represent relational data, so we would like to extend these planar languages to a class of languages allowing for relations, but retaining the learning convergence properties. To this end, in "*Tree Planar Languages*", Florencio introduces the class of tree planar languages, where the data can be described as a tree, which is then mapped to a point in higher-dimensional space, where learning occurs, and can then be mapped back, identifying the tree-based concept

learned. And, this formulation still retains the learning convergence properties of planar languages. While ultimately we hope to show similar results for graph planar languages, these results are promising for domains such as natural language processing, web mining, bioinformatics and computer vision.

Mining structural data usually involves either the classification of nodes or the prediction of links in the network. Another learning task is anomaly detection, and in the realm of relational learning, anomalies can take the form of relational variants. In "*Discovering Structural Anomalies in Graph-Based Data*", Eberle and Holder present methods for identifying anomalies in the structure of relational data represented as a graph. Their methods rely on a definition of anomaly as a small, unexpected deviation to a normative pattern. Such a definition is important for fraud detection, where the perpetrator attempts to mimic normal behavior. They evaluate their methods on synthetic data containing a prevalent pattern and then anomalies to the pattern. The results show that the methods have high accuracy at identifying the anomalies with low false positive rates. Their methods also perform well on two real-world tasks involving cargo smuggling and intrusion detection. With the data collected by various fraud-detection entities becoming increasingly relational, these methods represent the next step in incorporating relational information in the pursuit of fraudulent activity.

Frequent subgraph mining is one of the more prevalent graph mining tasks and seeks to identify all subgraphs that exist in some fraction of a set of graphs. One variant to this task is when the data consists of one large graph, rather than a set of graphs. This variant introduces a complication for determining the frequency of a subgraph, when the instances, or embeddings, of the subgraph overlap in the large graph. Two instances that overlap do not represent as much support for the subgraph as two independent instances. Yet, however we count the instances, we must ensure that the anti-monotone property of frequent subgraph mining (i.e., that supergraphs of a subgraph will have at most the same frequency as the subgraph) is maintained in order to preserve the performance gained by being able to prune extensions of a subgraph with less frequency. In their paper "*Subgraph Support in a Single Large Graph*", Fiedler and Borgelt address this issue by analyzing several methods for counting overlapping embeddings of a subgraph in one large graph. They find that while the methods all satisfy the anti-monotone property, they differ in the frequency counts for subgraphs. Therefore, frequent subgraph miners employing different counting methods may return different results for the same minimum support. Specifically, some overlapping embeddings can be considered harmless, in that counting them all will not violate the anti-monotone property and therefore increase the set of frequent subgraphs for a given minimum support level. They also provide a clear proof of the anti-monotonicity of the MIS-support proposed in previous work. These results improve our understanding of handling overlapping embeddings in frequent subgraph mining and may improve performance in certain domains by identifying frequent subgraphs missed by other methods.

3. KEYNOTE TALK

David Jensen from University of Massachusetts at Amherst gave us a keynote talk, titled " *Learning Causal Dependencies in Networks*". In his talk, David briefly surveyed recent work in learning probabilistic models of relational data, and discussed several applications of these techniques, including fraud detection in the U.S. securities industry. David argued that current techniques are capable of learning only a subset of the knowledge needed by practitioners in these domains, and that informing effective action often requires a causal model. He then addressed the open question of whether relational representations make the problem of learning causal models easier or harder, and presented some reasons for optimism that relational representations may be able to greatly improve our ability to learn such models.

In summary, this workshop has provided many attractive topics for further study in graph mining. Specifically, mining massive graphs becomes one of the main research themes. Around two thirds of papers presented in this workshop are related to this topic, which includes clustering, classification and pattern mining in massive graphs. It shows that graph mining becomes the must-have method for analyzing social networks, biological networks, the Web and relational data.

4. ACKNOWLEDGMENTS

MGCS 2007 would thank the program committee members for their contributions: Jiawei Han (UIUC), Yan Liu (IBM), Thomas Gaertner (Fraunhofer Inst. for Auto. Intel. Sys.), Michael R. Berthold (Univ. of Konstanz), Takashi Washio (Osaka Univ.), Frank Olken (NSF), Istvan Jonyer (Oklahoma State Univ.), Ehud Gudes (Ben-Gurion Univ.), Lise Getoor (Univ. of Maryland), Tina Eliassi-Rad (LLNL), Karsten M. Borgwardt (Univ. of Munich), Joost N. Kok (Leiden Univ.), Siegfried Nijssen (Katholieke Univ. Leuven), Thorsten Meinl (Univ. of Konstanz), Yun Chi (NEC Lab), Jason T-L Wang (NJST Univ.), Mohammed J. Zaki (PRI), and Christos Faloutsos (CMU).