**Name:**

**Asish Balu**

**Title:**

**Enhancement of Taxonomy Generation and Contrastive Pattern Analysis for Heterogenous Information Network Visualization Frameworks.**

**Abstract:**

Heterogeneous information networks provide a meaningful way to store large amounts of related data in the form of nodes and links. As datasets continue to increase in size and patterns become more complex and varied, the need for efficient storage and analysis of such networks has become paramount to their viability in research and industry. CubeNet is a novel framework for heterogeneous information networks that is analogous to a text-cube for text and phrase mining. CubeNet allows OLAP style selection and aggregation to allow users to view sub networks based on a subset of the overall data; smaller nodes are merged into more general “super-nodes” in order to improve processing efficiency and understandability. Although the system effectively organizes data from several large datasets, the framework can be enhanced to improve its organization, usefulness, and versatility. In particular, we explore enhancements to three key areas of the CubeNet framework: dimension/label generation, contrastive pattern analysis, and application extension. First, we look into designing a network version of common text mining label generation methods and attempt to solve dimension construction and cell construction simultaneously. Second, we enable contrastive pattern analysis within the CubeNet framework to find meaningful shared network patterns between adjacent cells. Finally, we enhance CubeNet’s functionality to work seamlessly with other datasets, thus extending its applications.

**Introduction:**

Datasets are getting larger every day. Social media networks, for instance, can consist of billions of nodes and trillions of links. Previous graph-network analysis methods cannot efficiently scale to such large networks without losing accuracy or accessibility. A company cannot make confident business decisions by analyzing network patterns consisting of millions or billions of nodes. For this reason, researchers have developed several methods to reduce or aggregate the data space used to build graph networks. Based on each user’s specific needs, can we consolidate the network structure into something more manageable yet more useful?

Following guidance from research in Text Mining, we can apply some of the same techniques to heterogeneous networks organize data. Analogous to TextCube but for networks, CubeNet uses the OLAP cube framework to store graph networks in multi-dimensional cells. Several key functionalities are supported, including basic OLAP style drill-down and roll-up procedures as well as query based graph network construction. In addition, CubeNet provides statistical analysis on generated subnetworks in addition to pattern finding across fixed dimensions. With this framework, the study of large heterogeneous graph networks is divided into smaller representative sub networks, thus providing maximum utility to the user.

CubeNet consists of three main parts: construction, analysis, and mining. In the construction section, several text mining techniques are used to extract quality phrases as well as links between these phrases. These phrases comprise the “nodes” of the system, and are linked to other nodes based on the extracted links. Using recent research on taxonomy generation techniques, a labels are created for each particular dimension of the data (ex. In the DBLP dataset, for the Author dimension some labels include Andrew Ng and David Blei). Note that in the current implementation of CubeNet, only 2-level taxonomies are supported. This means that each node has only two labels, one representing the dimension of the represented attribute, and one representing a sub-dimension. For instance, the node “web pages” has the parent label “application” which in turn has the parent label “phrase”, which represents the dimension of this node.

In the text mining world, the TaxoGen algorithm provides an efficient way to generate n-level taxonomies for a particular set of text data, as well as assign generated labels to existing nodes. The issue with TaxoGen is two-fold: First, the process is completed from a text mining perspective which may not extend perfectly to a heterogeneous graph network. Second, TaxoGen uses an unsupervised hierarchical clustering method that does not always produce satisfactory results. In order to improve taxonomy generation and labeling for CubeNet, new methods will have to be developed that take the network structure and links into account.

After construction of the CubeNet data cube, informative statistical measures such as subnetwork density, clustering coefficient, and subnetwork radius can be calculated for cells adjacent to the selected data subset. In particular, contrastive pattern analysis provides useful data on which patterns are shared across a particular dimension. For example, if the user requests a pattern analysis over the Author dimension of the DBLP dataset, then all other dimensions are fixed while significant patterns are displayed for each Author.

Currently, CubeNet locates significant patterns using a scoring system based on pattern popularity, distinctiveness, and integrity. At a general level, popularity is how frequently this pattern appears overall, distinctiveness is how often this pattern appears in this cell only, and integrity is how often this pattern appears compared to its sub-patterns. Under the right parameters, this scoring system may produce relevant patterns, but the thresholds for popularity, distinctiveness, and integrity may vary across datasets. This paper will look into other measures of evaluating frequent measures and implement these methods within the CubeNet framework.

Both construction and analysis of graph network patterns are performed with high efficiency and accuracy on the DBLP dataset; however, this functionality is not replicated across datasets. To guarantee CubeNet will work properly for future datasets, we look at ways of modifying the underlying architecture of the system. The primary goal is to make improvements to the system to ensure seamless visualization of datasets irrespective of their scale.

**Related Work:**

The labels used in the current implementation of CubeNet are based primarily on existing data from the web. In addition, some nodes were labeled through manual methods. Initially, TaxoGen was considered to generate taxonomies as well as assign labels to particular nodes.

TaxoGen creates n-level taxonomies through a top-down recursive process. First it clusters the input corpus into several sub topics. Then it pushes any non-representative terms to the parent topic. A reprsentative term is one that appears frequtntly in documents of that particular topic as well as one that appears much more frequently in that sub-topic compared to other sub-topics. If the number of sub-topics is K, then K+1 clusters are formed, with the (K+1) cluster denoting the parent cluster. The process is repeated with each of the sub-topic clusters until no more terms are pushed to the corresponding parent cluster. This indicates that the sub-topics only contain representative terms.

A key part of TaxoGen is that it uses local word embeddings instead of global word embeddings. Basically, word embeddings are re-learned in each cluster based on the other terms in the cluster, which ensures that the embeddings reflect the semantic meaning of the particular sub-topic they are associated with.

Unfortunately, TaxoGen fails to achieve high quality results for graph networks due to its lack of supervision. In addition, TaxoGen constructs term embeddings using general text mining methods (such as a local context window) and does not take into account network links, which could provide meaningful relationship information between nodes. Since TaxoGen is ineffective for graph networks, CubeNet opted for a weakly supervised labeling algorithm to classify nodes. This is essentially a graph-based semi-supervised learning problem.

In order to generate frequent patterns of adjacent cells, CubeNet leverages techniques from CloseGraph to efficiently mine closed network patterns.

CloseGraph is a method to mine closed frequent sub-graphs (networks) of a given graph network. It starts with the gSpan algorithm which discovers frequent patterns through clever edge growth methods. The issue with gSpan is that while it can find frequent patterns quite efficiently, it does not differentiate between closed and non-closed patterns. The computational effort required to filter through all the frequent patterns and only keep the closed patterns is enormous, so CloseGraph proposes a method to mine closed graphs directly. Essentially, CloseGraph prunes large portions of the search space by taking advantage of the fact that some edge patterns must occur together, which prevents many super-graphs from being closed.

Although CloseGraph can find frequent closed patterns efficiently, it does not take into account the contexts of the network. To make up for this, CubeNet employs a pattern quality measure that takes into account the popularity, distinctiveness, and integrity of mined patterns. While this quality measure works for some datasets, it does not extend accurately to other datasets.

**Reference List:**

**Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian Sadler, Michelle Vanni and Jiawei Han, "**[TaxoGen: Constructing Topical Concept Taxonomy by Adaptive Term Embedding and Clustering](http://hanj.cs.illinois.edu/pdf/kdd18_czhang.pdf)**", in Proc. of 2018 ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'18), London, UK, August 2018**

Fangbo Tao, Honglei Zhuang, Chi Wang Yu, Qi Wang, Taylor Cassidy, Lance Kaplan, Clare Voss, Jiawei Han, "[Multi-Dimensional, Phrase-Based Summarization in Text Cubes](http://hanj.cs.illinois.edu/pdf/deb16_ftao.pdf)", Data Eng. Bulletin 39(3), Sept. 2016, pp. 74-84

Xifeng Yan and Jiawei Han. 2003. CloseGraph: mining closed frequent graph paerns. In KDD. 286–295.

Carl Yang, Mengxiong Liu, Frank He, Xikun Zhang, Jian Peng, and Jiawei Han. 2018. Similarity Modeling on Heterogeneous Networks via Automatic Path Discovery. In ECML-PKDD. 37–54.