Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 67 (2014) 21705-21709

ISSN: 2229-712X

A survey on Efficient and Scalable method for Learning Collective Behavior

Umesh B.Shingote* and Setu Chaturvedi

Department of Computer Science & Engineering, Technocrats Institute of Technology, Bhopal, Madhya Pradesh, India.

ARTICLE INFO

Article history:

Received: 27 January 2014;

Received in revised form:

7 February 2014;

Accepted: 13 February 2014;

Keywords

Social networks,
Users Behavior,
Learning,
Prediction of behavior,
k-means variant,
Clustering.

ABSTRACT

For the many people in world social networks are playing major role day to day life. Depending on the user's behavior and interaction with each other, the social networking sites are reshaped. Growing interest and development of social network sites like Facebook, Twitter, Flicker, YouTube etc. imposing many research challenges. And hence this is allowing researchers to do many research studies using data mining concepts. The main challenge of such online social networking websites is to find out the individuals behavior over social network. Understanding the user's behavior on social networking websites is called as collective behavior. There are many data mining techniques presented to identify the behavior of individuals. Such methods of collective behavior allows to learn and predict the users online behavior and based on it assign the appropriate label to actor in network. But the another main problem occurs in such methods is the networks scalability due to which this systems becomes poor in performance and many not be work if the network size is too big. To overcome this problem we need to have scalable learning of collective behavior to deal with any size of social networks. Recently one such method presented, in this method an edge-centric clustering technique is presented to extract social network dimensions. With sparse social dimensions, the proposed approach can efficiently handle networks of any size. In this paper we are presenting the detailed discussion on this method.

Introduction

Social Media allowing more connectivity and interaction between web users and it encourages contributions and feedback from anyone who is a member of any virtual community. Social media will alter new mass cooperative behaviors that unlock the ability of the collective and deliver new ways to enterprise results. Enterprises will use these collective behaviors because the link between business worth and social media technologies. Enterprises will use them to look at a target community and formulate new ways in which folks will move to realize enterprise worth. In gift World Social media facilitate folks of all walks of life to attach to every different. We tend to study however networks in social media will facilitate predict some kinds of human behavior and individual preference. In social media, countless actors in an exceedingly network area unit the norm. With this large variety of actors, the scale cannot even be command in memory, inflicting significant issue concerning the quantifiability. As an example, standard content- sharing sites like Del.icio.us, Flickr and YouTube enable users to transfer, tag and comment differing types of contents (bookmarks, photos, videos).Users registered at these sites may become friends, an addict or follower of others. The prolific and swollen use of social media has flipped on-line interactions into a significant a part of human expertise.

In this work, we tend to study however networks in social media will facilitate predict some human behaviors and individual preferences. Above all, given the behavior of some people in an exceedingly network, how can we infer the behavior of other individuals in the same social network [1].This study will facilitate higher perceive activity patterns of users in social media for applications like social advertising and recommendation. A social-dimension-based approach has been shown effective in addressing the heterogeneity of connections presented in social media. But the networks in social media are normally of colossal size, involving hundreds of thousands of actors. The scale of these networks entails scalable learning of models for collective behavior prediction.

To address the scalability issue, new method introduced an edge-centric clustering scheme to extract sparse social dimensions [1].With sparse social dimensions, the proposed approach can efficiently handle networks of millions of actors while demonstrating a comparable prediction performance to other non-scalable methods. Social media facilitate people of all walks of life to connect to each other. In this paper we are presenting and discussing the all algorithms involved in this study with its practical evaluation.

Collective behavior refers to the behaviors of individuals in a social networking environment, but it is not simply the aggregation of individual behaviors. In a connected environment, individuals' behaviors tend to be interdependent, influenced by the behavior of friends. This leads to behavior correlation between connected users. Take marketing as an example: if our friends buy something, there is a better-than-average chance that we will buy it, too. This behavior correlation can also be explained by homophile. The recent boom of social media enables us to study collective behavior on a large scale. Here, behaviors include a broad range of actions: joining a group, connecting to a person, clicking on an ad, becoming interested in certain topics, dating people of a certain type, etc. In this work, we attempt to leverage the behavior correlation presented in a social network in order to predict collective behavior in social media. In next section sections we will discuss the algorithms those are presented to overcome the limitations of existing methods by improving the network scalability. Examples of Behavior, Joining a sports club, buying some products, Becoming interested in a topic voting for a presidential candidate.

Tele:
E-mail addresses: umeshshingote62@gmail.com

Material and Methodology

There are many methods presented to learn and predict the collective behavior of users over online social networks. In [1], author presented all the related works and methods for learning collective behavior. Solution for the problem of scalability by using the new data mining methods is provided. Below are some of the methods presented previously [1].Classifications with networked instances are known as within-network classification, or a special case of relational learning [2, 3]. But network tends to present heterogeneous relations, and the Markov assumption can only capture the local dependency. In [4, 5], there is model network connections or class labels based on latent groups. Similar idea is also adopted to differentiate heterogeneous relations in a network by extracting social dimensions to represent the potential affiliations of actors in a network [6].

Modularity Maximization (ModMax)

Some researchers presented methods to conduct soft clustering for graphs like modularity maximization [8]. Probabilistic methods are also developed [9, 10]. Special variant of modularity maximization is adopted to extract social dimension. Social dimension corresponds to the top eigenvectors of a modularity matrix. A disadvantage with soft clustering such as modularity maximization is that the resultant social dimensions are dense, posing thorny computational challenges for the extraction of extraction of social dimensions and discriminative learning. The modularity maximization requires the computation of the top eigenvectors of a modularity matrix which is of size $n \times n$ where n is the number of actors in a network. When the network scales to millions of actors, the eigenvector computation becomes a daunting task. Though the network is sparse, the social dimensions become dense. Efficient online updates of Eigen vectors with expanding matrices remain a challenge. It destroys the genuine sparsity presented in a network.

Advantages

- Use top eigenvectors of a modularity matrix as social dimensions.
- Outperform representative methods based on collective inference.

Disadvantage

- Dense Representation. e.g. 1 M actors, 1000 dimensions, require 8G memory.
- Eigenvector computation can be expensive.
- Difficult to update whenever the network changes.

Bi–connected Components (BiComponents)

Another related approach to finding edge partitions is bi-connected components [11]. Bi-connected components of a graph are the maximal subsets of vertices such that the removal of a vertex from a particular component will not disconnect the component. Essentially, any two nodes in a bi-connected component are connected by at least two paths. It is highly related to cut vertices (a.k.a. articulation points) in a graph, whose removal will result in an increase in the number of connected components. Those cut vertices are the bridges connecting different bi-connected components. Thus, searching for bi-connected components boils down to searching for articulation points in the graph, this can be solved efficiently in O (n + m) time. Here n and m represent the number of vertices and edges in a graph, respectively. Each bi-connected component is considered a community, and converted into one social dimension for learning.

Advantages:

- BiComponents separates edges into disjoint sets which in turn deliver sparse social dimension.
- BiComponents is very efficient and scalable.

Disadvantage:

- BiComponents gives output highly imbalanced communities.
- It fails to extract informative social dimension for classification.
- This technique yields poor performance.

Node Clustering (NodeCluster)

Note that social dimensions allow one actor to be involved in multiple affiliations. As a proof of concept, we also examine the case when each actor is associated with only one affiliation. Essentially, we construct social dimensions based on node partition. A similar idea has been adopted in a latent group model [21] for efficient inference. To be fair, we adopt k-means clustering to partition nodes of a network into disjoint sets, and convert the node clustering result into a set of social dimensions. Then, SVM is utilized for discriminative learning. For convenience, this method is denoted as NodeCluster.

Advantage

- SVM is used for discriminative learning.

Disadvantage

- Each actor to be involved in only one affiliation, yielding in inferior performance than EdgeCluster.

Edge Clustering (EdgeCluster)

Consequently, it is imperative to develop scalable Methods that can handle large-scale networks efficiently without extensive memory requirements. Next, we elucidate on an edge-centric clustering scheme to extract sparse social dimensions. With such a scheme, we can also update the social dimensions efficiently when new nodes or new edges arrive. In a huge network, large number of social dimension needs to be extracted. Apply k-means algorithm to partition edges into disjoint sets.

Space: O (n+m) Time: O (m)

Advantage

- EdgeCluster method extract social dimension which are sparse. (Theoretically Guaranteed)
- One actor can be assigned to multiple affiliations
- Easy to update with new edges and nodes
- Simply update the centroid.

We have studied that proposed approach of edge- centric view for social dimension extraction and liner SVM based approach for learning of collective behavior. From the experimental results it’s clear that proposed social dimension extraction technique outperforms existing social dimension extraction techniques [1].

In addition to this, one more advantage is that this model is that it easily scales to handle networks with millions of actors while the earlier models fail. However, generally in all social networks, multiple nodes of actors are involving in similar network and hence this is resulting as multimode network. For example For instance, in Flicker, photos, users, communities, comments etc. in YouTube, users, videos, tags, and comments are twist together in co-existence. Such heterogeneity of social networks needs to be handled during the edge-centric clustering so that we can improve the prediction performance especially in case of multimode networks. In this section we will discuss the recently scalable approach for learning collective behavior.

Social Dimension Extraction using K- means variant algorithm

In this algorithm we design the framework, design the input datasets, and implement the algorithm of k-means variant in order to extract the social dimension extraction based edge centric approach. As a simple k-means is adopted to extract social dimensions, it is easy to update social dimensions if a given network changes.

Input: Social Network Dataset like flicker, YouTube.

Output: Social dimensions of this dataset. For example, following figures showing Input and Output of this module:

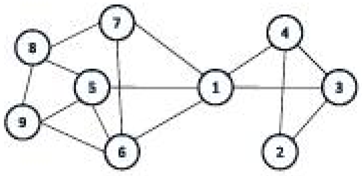

Figure 1. A toy example

Given a network (say, Figure 1), we take the Edge-centric view of the network data (Table 1 below):

Table 1. Edge Centric view of input network data

Edge	Features								
	1	2	3	4	5	6	7	8	9
$e(1,3)$	1	0	1	0	0	0	0	0	0
$e(1,4)$	1	0	0	1	0	0	0	0	0
$e(2,3)$	0	1	1	0	0	0	0	0	0
⋮	....								

Further partition this information (edges) into the disjoint sets as per given in below figure 2:

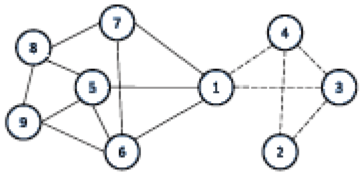

Figure 2. Edge Cluster

Based on Edge clustering above, further social dimensions can be constructed as per given in below table 2 and this is the final output of this algorithm 1:

Table 2. Social Dimension(s) of the Toy Example

Actor	Modularity Maximization	Edge Partition	
1	-0.1185	1	1
2	-0.4043	1	0
3	-0.4473	1	0
4	-0.4473	1	0
5	0.3093	0	1
6	0.2628	0	1
7	0.1690	0	1
8	0.3241	0	1
9	0.3522	0	1

Discriminative Learning and Prediction

We have applied the social extracted dimension input to the discriminative learning and prediction. We have to use algorithm given in [1] for this section. This algorithm is based on linear SVM.

Input: network data, labels of some nodes, number of social dimensions;

Output: labels of unlabeled nodes.

Step 1: Apply regularization to social dimensions.

Step 2: Construct classifier based on social dimensions of labeled nodes.

Step 3: Use the classifier to predict labels of unlabeled ones based on their social dimensions.

Comparative Results Analysis

This prediction problem is essentially Multi-label. It is empirically shown that thresholding can affect the final prediction performance drastically [18, 19]. For evaluation purpose, we assume the number of labels of unobserved nodes is already known, and check whether the top-ranking predicted labels match with the actual labels. Such a scheme has been adopted for other multi-label evaluation works [20]. We randomly sample a portion of nodes as labeled and report the average performance of 10 runs in terms of Micro-F1 and Macro-F1 [19].

In this section, we first examine how prediction performances vary with social dimensions extracted following different approaches. Then we verify the sparsity of social dimensions and its implication for scalability. We also study how the performance varies with dimensionality. Finally, concrete examples of extracted social dimensions are given. From all practical analysis this proposed methods showing good performances against the existing methods as well as improves scalability as well.

Prediction Performance

Prediction Performance on all data is shown in Tables 3-5. The entries in bold face denote the best performance in each column. Obviously, EdgeCluster is the winner most of the time. Edge-centric clustering shows comparable performance to modularity maximization on Blog Catalog network, yet it outperforms ModMax on Flickr. ModMax on YouTube is not applicable due to the scalability constraint. Clearly, with sparse social dimensions, we are able to achieve comparable performance as that of dense social dimensions. We note that the prediction performance on the studied social media data is around 20-30% for F1 measure. This is partly due to the large number of distinctive labels in the data. Another reason is that only the network information is exploited here.

Observations

(ModMax-500 corresponds to modularity maximization to select 500 social dimensions and EdgeCluster-x denotes edge-centric clustering to construct x dimensions. Time denotes the total time (seconds) to extract the social dimensions; Space represents the memory footprint (mega-byte) of the extracted social dimensions; Density is the proportion of non-zeros entries in the dimensions; Upper bound is the density upper bound computed. Max-Aff and Ave-Aff denote the maximum and average number of affiliations one user is involved in).

The social dimensions constructed according to edge-centric clustering are guaranteed to be sparse because the density is upper bounded by a small value. Here, we examine how sparse the social dimensions are in practice. We also study how the computation time (with a Core2Duo E8400 CPU and 4GB memory) varies with the number of edge clusters. The computation time, the memory footprint of social dimensions, their density and other related statistics on all three data sets are reported in Tables 6-8.

Concerning the time complexity, it is interesting that computing the top eigenvectors of a modularity matrix is actually quite efficient as long as there is no memory concern. This is observed on the Flickr data. However, when the network scales to millions of nodes (YouTube), modularity maximization becomes difficult (though an iterative method or distributed computation can be used) due to its excessive memory requirement. On the contrary, the EdgeCluster method can still work efficiently as shown in Table 8. The computation time of EdgeCluster for YouTube is much smaller than for Flickr, because the YouTube network is extremely sparse. The number of edges and the average degree in YouTube are smaller than those in Flickr.

Table 3. Performance on Blog Catalog Network

Proportion of Labeled Nodes		10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1 (%)	EdgeCluster	27.94	30.76	31.85	32.99	34.12	35.00	34.63	35.99	36.29
	BiComponents	16.54	16.59	16.67	16.83	17.21	17.26	17.04	17.76	17.61
	ModMax	27.35	30.74	31.77	32.97	34.09	36.13	36.08	37.23	38.18
	NodeCluster	18.29	19.14	20.01	19.80	20.81	20.86	20.53	20.74	20.78
Macro-F1 (%)	EdgeCluster	16.16	19.16	20.48	22.00	23.00	23.64	23.82	24.61	24.92
	BiComponents	2.77	2.80	2.82	3.01	3.13	3.29	3.25	3.16	3.37
	ModMax	17.36	20.00	20.80	21.85	22.65	23.41	23.89	24.20	24.97
	NodeCluster	7.38	7.02	7.27	6.85	7.57	7.27	6.88	7.04	6.83

Table 4. Performance on Flickr Network

Proportion of Labeled Nodes		1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
Micro-F1 (%)	EdgeCluster	25.75	28.53	29.14	30.31	30.85	31.53	31.75	31.76	32.19	32.84
	BiComponents	16.45	16.46	16.45	16.49	16.49	16.49	16.49	16.48	16.55	16.55
	ModMax	22.75	25.29	27.30	27.60	28.05	29.33	29.43	28.89	29.17	29.20
	NodeCluster	22.94	24.09	25.42	26.43	27.53	28.18	28.32	28.58	28.70	28.93
Macro-F1 (%)	EdgeCluster	10.52	14.10	15.91	16.72	18.01	18.54	19.54	20.18	20.78	20.85
	BiComponents	0.45	0.46	0.45	0.46	0.46	0.46	0.46	0.46	0.47	0.47
	ModMax	10.21	13.37	15.24	15.11	16.14	16.64	17.02	17.10	17.14	17.12
	NodeCluster	7.90	9.99	11.42	11.10	12.33	12.29	12.58	13.26	12.79	12.77

Table 5. Performance on YouTube Network

Proportion of Labeled Nodes		1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
Micro-F1 (%)	EdgeCluster	23.90	31.68	35.53	36.76	37.81	38.63	38.94	39.46	39.92	40.07
	BiComponents	23.90	24.51	24.80	25.39	25.20	25.42	25.24	24.44	25.62	25.53
	ModMax	-	-	-	-	-	-	-	-	-	-
	NodeCluster	20.89	24.57	26.91	28.65	29.56	30.72	31.15	31.85	32.29	32.67
Macro-F1 (%)	EdgeCluster	19.48	25.01	28.15	29.17	29.82	30.65	30.75	31.23	31.45	31.54
	BiComponents	6.80	7.05	7.19	7.44	7.48	7.58	7.61	7.63	7.76	7.91
	ModMax	-	-	-	-	-	-	-	-	-	-
	NodeCluster	17.91	21.11	22.38	23.91	24.47	25.26	25.50	26.02	26.44	26.68

Table 6. Sparsity Comparison on Blog Catalog data with 10,312 Nodes

Methods	**Time**	**Space**	**Density**	**Upper Bound**	**Max-Aff**	**Ave Aff**
ModMax 500	194.4	41.2M	1	-	500	500
EdgeCluster -100	300.8	3.8M	1.1×10^{-1}	2.2×10^{-1}	187	23.5
EdgeCluster -500	357.8	4.9M	6.0×10^{-2}	1.1×10^{-1}	344	30.0
EdgeCluster -1000	307.2	5.2M	3.2×10^{-2}	6.0×10^{-2}	408	31.8
EdgeCluster -2000	294.6	5.3M	1.6×10^{-2}	3.1×10^{-2}	598	32.4
EdgeCluster -5000	230.3	5.5M	6×10^{-3}	1.3×10^{-2}	682	32.4
EdgeCluster -10000	195.6	5.6M	3×10^{-3}	7×10^{-3}	882	33.3

Table 7. Sparsity Comparison on Flickr Data with 80, 513 Nodes

Methods	**Time**	**Space**	**Density**	**Upper Bound**	**Max-Aff**	**Ave Aff**
ModMax 500	2.2×10^{3}	322.1M	1	-	500	500.0
EdgeCluster -200	1.2×10^{4}	31.0M	1.2×10^{-1}	3.9×10^{-1}	156	24.1
EdgeCluster -500	1.3×10^{4}	44.8M	7.0×10^{-2}	2.2×10^{-1}	352	34.8
EdgeCluster -1000	1.6×10^{4}	57.3M	4.5×10^{-2}	1.3×10^{-1}	619	44.5
EdgeCluster -2000	2.2×10^{4}	70.1M	2.7×10^{-2}	7.2×10^{-1}	986	54.4
EdgeCluster -5000	2.6×10^{4}	84.7M	1.3×10^{-2}	2.9×10^{-2}	1405	65.7
EdgeCluster -10000	1.9×10^{4}	91.4M	7×10^{-3}	1.5×10^{-2}	1673	70.9

Table 8. Sparsity Comparison on YouTube Data with 1, 138,499 Nodes

Methods	**Time**	**Space**	**Density**	**Upper Bound**	**Max-Aff**	**Ave Aff**
ModMax 500	N/A	4.5G	1	-	500	500.00
EdgeCluster -200	574.7	36.2M	9.9×10^{-3}	2.3×10^{-2}	121	1.99
EdgeCluster -500	606.6	39.9M	4.4×10^{-3}	9.7×10^{-3}	255	2.19
EdgeCluster -1000	779.2	42.3M	2.3×10^{-3}	5.0×10^{-3}	325	2.32
EdgeCluster -2000	558.9	44.2M	1.2×10^{-3}	2.6×10^{-3}	375	2.43
EdgeCluster -5000	554.9	45.6M	5.0×10^{-4}	1.0×10^{-3}	253	2.50
EdgeCluster -10000	561.2	46.4M	2.5×10^{-4}	5.1×10^{-4}	356	2.54
EdgeCluster -20000	507.5	47.0M	1.3×10^{-4}	2.6×10^{-4}	305	2.58
EdgeCluster -50000	597.4	48.2M	5.2×10^{-5}	1.1×10^{-4}	297	2.62

Another observation is that the computation time of EdgeCluster does not change much with varying numbers of clusters. No matter how many clusters exist, the computation time of EdgeCluster is of the same order. This is due to the efficacy of the proposed k-means variant. In the algorithm, we do not iterate over each cluster and each centroid to do the cluster assignment, but exploit the Sparsity of edge-centric data to compute only the similarity of a centroid and those relevant instances. This, in effect, makes the computational cost independent of the number of edge clusters. As for the memory footprint reduction, sparse social dimension does an excellent job. On Flickr, with only500 dimensions, ModMax requires 322.1M, whereas EdgeCluster requires less than 100M. This effect is stronger on the mega-scale YouTube network, where ModMax becomes impractical to compute directly. Itis expected that the social dimensions of ModMax would occupy 4.6G memory. On the contrary, the sparse social dimensions based on EdgeCluster require only 30-50M. The steep reduction of memory footprint can be explained by the density of the extracted dimensions.

For instance, in Table 8, when we have 50,000 dimensions, the density is only 5.2×10^{-5}. Consequently, even if the network has more than 1 million nodes, the extracted social dimensions still occupy only a tiny memory space. The upper bound of the density is not tight when the number of clusters k is small. As k Increases, the bound becomes tight. In general, the true density is roughly half of the estimated bound.

Conclusion and Future Work

In this paper we have studied efficient method for collective behavior. This method is presented to address the issues of scalability of all existing methods.

(i) This method is presented to address the issues of scalability using an edge-centric clustering scheme to extract social dimensions and a scalable k-means variant to handle edge clustering.

(ii) From the experimental studies it shows comparable prediction performance as earlier proposed approaches to extract social dimensions

(iii) In actual, each edge can be associated with multiple affiliations while our current model assumes only one dominant affiliation.

(iv) This investigated method is sensitive to the number of social dimensions. The advantage of this method is that it easily scales to handle networks with millions of actors while existing methods was failed to do so. This scalable approach offers a viable solution to effective learning of online collective behavior on a large scale. However as per stated in limitations, this method further needs to improve in different directions. In social media, multiple modes of actors can be involved in the same network, resulting in a multimode network [17]. Extending the edge-centric clustering scheme to address this object heterogeneity can be a promising future direction. Since the proposed EdgeCluster model is sensitive to the number of Social dimensions as shown in the experiment, further research are needed to determine a suitable dimensionality automatically. Since the proposed EdgeCluster model is sensitive to the number of social dimensions as shown in the experiment, further research is needed to determine a suitable dimensionality automatically. It is also interesting to mine other behavioral features (e.g., user activities and temporal spatial information) from social media, and integrate them with social networking information to improve prediction performance.

References

[1] L. Tang, H. Liu, "Toward predicting collective behavior via social dimension extraction," IEEE Intelligent Systems, vol. 25, pp. 19–25, 2010

[2] S. A. Macskassy and F. Provost, "Classification in networked data: A toolkit and a univariate case study", J. Mach. Learn. Res., Volume. 8, pp. 935–983, 2007.

[3] L. Getoor and B. Taskar, Eds., "Introduction to Statistical Relational Learning". The MIT Press, 2007.

[4] J. Neville and D. Jensen, "Leveraging relational autocorrelation with latent group models", in MRDM '05: Proceedings of the 4th international workshop on Multi-relational mining. New York, NY, USA: ACM, 2005, pp. 49–55.

[5] Z. Xu, V. Tresp, S. Yu, and K. Yu, "Nonparametric relational learning for social network analysis", in KDD'2008 Workshop on Social Network Mining and Analysis, 2008.

[6] "Relational learning via latent social dimensions in KDD '09": Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. New York, NY, USA: ACM, 2009, pp. 817–826.

[7] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," Physical Review E, Volume. 69,p. 026113, 2004.

[8] K. Yu, S. Yu, and V. Tresp, "Soft clustering on graphs", in NIPS, 2005.

[9] E. Airodi, D. Blei, S. Fienberg, and E. P. Xing, Mixed membership stochastic blockmodels,J. Mach. Learn. Res., Volume. 9, pp. 1981–2014, 2008.

[10] G. Palla, I. Der´enyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society", Nature, Volume.435, pp. 814–818, 2005.

[11] Y.Y.Ahn, J.P.Bagrow, S.Lehmann, "Link communities reveal multi-scale complexity in networks", 2009.

[12] M. Newman, "Finding community structure in networks using the eigenvectors of matrices," Physical Review E(Statistical, Nonlinear, and Soft Matter Physics), vol. 74,no. 3, 2006.

[13] T. Evans and R. Lambiotte, "Line graphs, link partitions, and overlapping communities", Physical Review E, Volume. 80, no. 1, pp. 16105, 2009.

[14] J. Hopcroft and R. Tarjan, "Algorithm 447: efficient algorithms for graph Manipulation," Commun. ACM, Volume 16, no. 6, pp. 372–378, 1973.

[15] L. Tang, H. Liu, J. Zhang, and Z. Nazeri, "Community evolution in dynamic multi-mode networks," in KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. New York, NY, USA: ACM, 2008, pp. 677–685.

[16] R.-E. Fan and C.-J. Lin, "A study on threshold selection for multi-label classification," 2007.

[17] L. Tang, S. Rajan, and V. K. Narayanan, "Large scale multilabel classification via metalabeler," in WWW '09: Proceedings of the 18th international conference on World Wide Web. New York, NY, USA: ACM, 2009, pp. 211–220.

[18] Y. Liu, R. Jin, and L. Yang, "Semi-supervised multi-label learning by constrained non-negative matrix factorization," in AAAI, 2006.

[19] J. Neville and D. Jensen, "Leveraging relational autocorrelation with latent group models," in MRDM '05: Proceedings of the 4thinternational workshop on Multi relational mining. New York, NY, USA: ACM, 2005, pp. 49–55.