International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-1 Issue-3, August 2012

Behavior Prediction Via Social Dimensions Extraction

M. Nagendramma, K. Subba Reddy

Abstract: ***Online social networks play an important role in everyday life for many people. Social media has reshaped the way in which people interact with each other. The rapid development of participatory web and social networking sites like YouTube, Twitter, and Face book also brings about many data mining opportunities and novel challenges. In particular, given information about some individuals, how can we infer the behavior of unobserved individuals in the same network? A social-dimension-based approach has been shown effective in addressing the heterogeneity of connections presented in social media. However, 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, we propose an edge-centric clustering scheme to extract sparse social dimensions. 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.***

Index Terms: ***Classification with network Data, Collective Behavior, community detection, social dimensions.***

I. INTRODUCTION

Recently, Social media like Face book and YouTube are becoming increasingly popular. But how to monetize the rocketing online traffic in social media is a big challenge. Unfortunately, in normal social networking sites, not like search engines, very limited user profile or intention information are available. Given the social network information, is it possible to infer the user preference or potential behavior?

In this work, we study how networks in social media can help predict some human behaviors and individual preferences. In particular, given the behavior of some individuals in a network, how can we infer the behavior of other individuals in the same social network [1]? This study can help better understand behavioral patterns of users in social media for applications like social advertising and recommendation.

In social media, the connections of the same network are not homogeneous. However, this relation type information is not readily available in reality. A framework based on social dimensions [2] is proposed to address this heterogeneity. In the initial study, modularity maximization [3] is exploited to extract social dimensions. With huge number of actors, the dimensions cannot even be held in memory.

Revised Manuscript Received on 30 August 2012.
* Correspondence Author
M.Nagendramma*, Department of CSE, prakasam Engineering college, kandukur India.
Prof. K.Subbareddy, Department of CSE, prakasam Engineering college, kandukur India.

In this work, we *propose an effective edge-centric* approach to extract sparse social dimensions. In social media, a network of millions of actors is very common. With a huge number of actors, extracted dense social dimensions cannot even be held in memory, causing a serious computational problem. Scarifying social dimensions can be effective in eliminating the *scalability* bottleneck. In this work, we propose an effective *edge-centric* approach to extract sparse social dimensions [4]. We prove that with our proposed approach, sparsity of social dimensions is guaranteed. Extensive experiments are then conducted with social media data. The framework based on sparse social dimensions, without sacrificing the prediction performance, is capable of efficiently handling real-world networks of millions of actors.

Procedure for Paper Submission

A. Review Stage

Submit your manuscript electronically for review.

B. Final Stage

When you submit your final version, after your paper has been accepted, prepare it in two-column format, including figures and tables.

C. Figures

As said, to insert images in *Word*, position the cursor at the insertion point and either use Insert | Picture | From File or copy the image to the Windows clipboard and then Edit | Paste Special | Picture (with “Float over text” unchecked).

The authors of the accepted manuscripts will be given a copyright form and the form should accompany your final submission.

II. COLLECTIVE BEHAVIOUR

When people are exposed in a social network environment, their behaviors can be influenced by the behaviors of their friends. People are more likely to connect to others sharing certain similarity with them. This naturally leads to behavior correlation between connected users [5].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 [6].

Given a network with the behavioral information of some actors, how can we infer the behavioral outcome of the remaining actors within the same network? Here, we assume the studied behavior of one actor can be described with K class labels $\{c1, \cdots, c_K\}$. Each label, ci , can be 0 or 1. For instance, one user might join multiple groups of interest, so

ci= 1 denotes that the user subscribes to group i, and ci = 0 otherwise. Likewise, a user can be interested in several topic simultaneously, or click on multiple types of ads.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

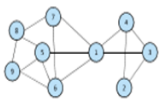
One special case is K = 1, indicating that the studied behavior can be described by a single label with 1and 0. For example, if the event is the presidential election, 1 or 0 indicates whether or not a voter voted for Barrack Obama .The problem we study can described formally as follows. Suppose there are K class labels Y ={c1 , · · · , c_K }. Given network G = (V, E, Y) where V is the vertex set, E is the edge set and Yi ⊆ Y are the class labels of a vertex $v_i \in V$, and known values of Yi for some subsets of vertices V^L , how can we infer the values of Yi (or an estimated probability over each label) for the remaining vertices $V^U = V - V^L$.

III. SOCIAL DIMENSIONS

Connections in social media are not homogeneous. People can connect to their family, colleagues, college classmates, or buddies met online. Some relations are helpful in determining a targeted behavior (category)while others are not. This relation-type information, however, is often not readily available in social media. A direct application of collective inference [8] or label propagation [9] would treat connections in a social network as if they were homogeneous. To address the heterogeneity present in connections, a framework (SocioDim) [2] has been proposed for collective behavior learning. The framework SocioDim is composed of two steps:1) social dimension extraction, and 2) discriminative learning. In the first step, latent social dimensions are extracted based on network topology to capture the potential affiliations of actors. These extracted social dimensions represent how each actor is involved in diverse affiliations.

In Existing approach the social dimensions extracted based on modularity maximization are the top eigenvectors of a modularity matrix. Though the network is sparse, the social dimensions become dense, require more memory space. E.g. 1 M actors, 1000 dimensions, requires 8G memory. Eigenvector computation can be expensive Difficult to update whenever the network changes Need a scalable algorithm to find sparse social dimensions.Let's look at the toy network in Figure 1. The column of modularity maximization in Table 1 shows the top eigenvector of the modularity matrix. Clearly, none of the entries is zero. This becomes a serious problem when the network expands into millions of actors and a reasonable large number of social dimensions need to be extracted. The eigenvector computation is impractical in this case.

Table 1: Social Dimension(s) of the Toy Example



Actors	Modularity Maximization	Partition	Edge
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

Fig 1: represents toy example

IV. ALGORITHM—EDGECLUSTER

In this section, we first study how the Edge-centric views and k-means variant is used to extract sparse social dimensions and learning of collective behavior.

A. *Edge-Centric View*

The *proposed scalable* algorithm is an *edge-centric view*, i.e., partitioning the edges into disjoint sets such that each set represents one latent affiliation. For instance, we can treat each edge in the toy network in Figure 2 as one instance, and the nodes that define edges as features. This results in a typical feature-based data format as in figure 2 table. Based on the features (connected nodes) of each edge, we can cluster the edges into two sets as in Figure 2, where the dashed edges represent one affiliation, and the remaining edges denote another affiliation. One actor is considered associated with one affiliation as long as any of his connections is assigned to that affiliation. Hence, the disjoint edge clusters in Figure 2 can be converted into the social dimensions as the last two columns for edge-centric clustering in Table 1. Actor 1 is involved in both affiliations under this *Edge Cluster* scheme.

- Apply k-means algorithm to partition edges into disjoint sets

1. One actor can be assigned to multiple affiliations
2. Sparse (Theoretically Guaranteed)
3. Scalable via k-means variant
   - Space: O(n+m)
   - Time: O(m)
4. Easy to update with new edges and nodes
   - Simply update the centroids

Fig.3 Overview of EdgeCluster Algorithm

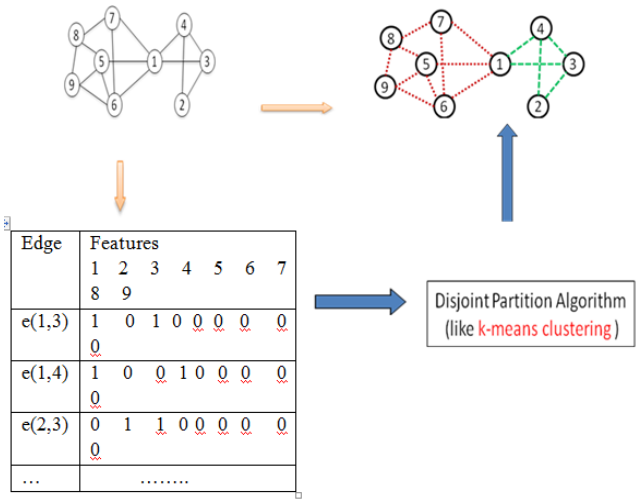


Fig.2 over view of edge cluster

In addition, the extracted social dimensions following edge partition are *guaranteed to be sparse*. This is because the number of one's affiliations is no more than that of her connections. Given a network with *m* edges and *n* nodes, if *k* social dimensions are extracted, then each node *vi* has no more than min(*di*, *k*) non-zero entries in her social dimensions, where *di* is the degree of node *vi*. We have the following theorem about the density of extracted social dimensions.

Theorem 1: Suppose *k* social dimensions are extracted from a network with *m* edges and *n* nodes. The density (proportion of nonzero entries) of the social dimensions based on edge partition is bounded by the following:

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

$$density \leq \frac{\sum_{i=1}^{n} min(d_{i,k})}{nk} \quad \text{.....(1)}$$

$$\frac{\sum_{\{i|d_i \leq k\}} d_i \quad + \sum_{\{i|d_i > k\}} k}{nk}$$

Moreover, for many real-world networks whose node degree follows a power law distribution, the upper bound in Eq. (1) can be approximated as follows:

$$\frac{\alpha-1}{\alpha-2}\frac{1}{k} - \left(\frac{\alpha-1}{\alpha-2} - 1\right) k^{-\alpha+1} \quad \text{..... (2)}$$

Where $\alpha > 2$ is the exponent of the power law distribution.

B. K-means Variant

As mentioned above, edge-centric clustering essentially treats each edge as one data instance with its ending nodes being features. Then a typical *k-means clustering* algorithm can be applied to find out disjoint partitions. One concern with this scheme is that the total number of edges might be too huge. Owning to the power law distribution of node degrees presented in social networks, the total number of edges is normally linear, rather than square, with respect to the number of nodes in the network.

By taking into account the two concerns above, we devise a k-means variant as shown in Figure 4

Input: data instances $\{x_i | 1 \leq i \leq m\}$

Number of clusters k

Output: $\{id\, x_i\}$

1. Construct a mapping from features to instances
2. Initialize the centroid of cluster {Cj |1 ≤ j ≤ k}
3. Repeat
4. Reset {M axSimi}, {idxi}
5. For j=1: k
6. Identify relevant instances Sj to centroid Cj
7. for i in Sj
8. Compute sim (i, Cj) of instance i and Cj
9. If sim (i, Cj) > MaxSimi
10. MaxSimi = sim (i, Cj)
11. idxi = j;
12. for i=1:m
13. update centroid C_{idxi}
14. until change of objective value < ϵ

Fig. 4. Algorithm of Scalable k-means Variant

By taking advantage of the feature-instance mapping, the cluster assignment for all instances (lines 5-11 in Figure 4) can be fulfilled in $O(m)$ time. Computing the new centroid (lines 12-13) costs $O(m)$ time as well. Hence, each iteration costs $O(m)$ time only. Moreover, the algorithm requires only the feature-instance mapping and network data to reside in main memory, which costs $O(m + n)$ space. Thus, as long as the network data can be held in memory, this clustering algorithm is able to partition its edges into disjoint sets. As a simple k-means is adopted to extract social dimensions, it is easy to update social dimensions if a given network changes. If a new member joins the network and a new connection emerges, we can simply assign the new edge to the corresponding clusters. The update of centroids with the new arrival of connections is also straightforward. This k-means scheme is especially applicable for dynamic large scale networks.

Hence by using the above described algorithms i.e Edge-cluster and k-means variant we can learn the collective behavior. Therefore the collective behavior algorithm shown in fig 5.

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

1. convert network into edge-centric view.
2. perform edge clustering as in Figure 4.
3. construct social dimensions based on edge partition. A node belongs to one community as long as any of its neighboring edges is in that community.
4. apply regularization to social dimensions.
5. construct classifier based on social dimensions of labeled nodes.
6. use the classifier to predict labels of unlabeled ones based on their social dimensions

Fig. 5. Algorithm for Learning of Collective Behavior

V. EXPERIMENT RESULTS

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

A. Prediction Performance

The *Edge Cluster* 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 Flicker. *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.

TABLE 2 Scarcity Comparison on Blog Catalog data with 10, 312 Nodes. ModMax-500 corresponds to modularity maximization to select 500 social dimensions and Edge Cluster-x denotes edge-centric clustering to construct x dimensions. Time denotes the total time (seconds) to extract the social dimensions; Space represent 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 following Eq. (1); Max-Aff and Ave-Aff denote the maximum and average number of affiliations one user is involved in.

Methods	Time	Space	Density	Upper Bound	Max-Aff	Ave-Aff
ModMax-500	194.4	41.2M	1	—	500	500
Edge Cluster – 100	300.8	3.8M	1.1×10^{-1}	2.2×10^{-1}	187	23.5
Edge Cluster – 500	357.8	4.9M	6.0×10^{-2}	1.1×10^{-1}	344	30.0
Edge Cluster – 1000	307.2	5.2M	3.2×10^{-2}	6.0×10^{-2}	408	31.8
Edge Cluster – 2000	294.6	5.3M	1.6×10^{-2}	3.1×10^{-2}	598	32.4
Edge Cluster – 5000	290.3	5.5M	6×10^{-3}	1.3×10^{-2}	682	32.4
edgeCluster – 10000	195.6	5.6M	3×10^{-3}	$7 \times 10^{-}$	882	33.3

Table 3 Sparsity Comparison on Flickr Data with 80, 513 Nodes

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Methods	Time	Space	Density	Upper Bound	Max-Aff	AveAff
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} 44.5	57.3M	4.5×10^{-2}	1.3×10^{-1}	619	
EdgeCluster-2000	2.2×10^{4}	70.1M	2.7×10^{-2}	7.2×10^{-2}	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 4 Sparsity Comparison on YouTube Data with 1, 138, 499 Nodes

Methods	Time	Space	Density	Upper Bound	Max-Aff	Ave-Aff
ModMax-500	N/A	4.6G	1	—	500	500.00
Edge Cluster-200	574.7	36.2M	9.9×10^{-3}	2.3×10^{-2}	121	1.99
Edge Cluster-500	606.6	39.9M	4.4×10^{-3}	9.7×10^{-3}	255	2.19
Edge Cluster-1000	779.2	42.3M	2.3×10^{-3}	5.0×10^{-3}	325	2.82
Edge Cluster-2000	558.9	44.2M	1.2×10^{-3}	2.6×10^{-3}	375	2.43
Edge Cluster-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}	366	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

B. Scalability Study

As we have introduced in Theorem 1, 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 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 2-4. 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 *Edge Cluster* method can still work efficiently as shown in Table 4. This is due to the efficacy of the proposed k-means variant in Figure 4. 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.

C. Chart Generation for User/Group

Two data sets reports are used to examine our proposed model for collective behavior learning. The first data set is acquired from user interest, the second from concerning behavior; we study whether or not a user visits a group of interest, then generates chart based on the user visit group in the month. The below chart contains communities Vs users. According to chart the commercial community contain more users than other so depends on this we can identify the user behavior i.e which type of features the users are attracted.

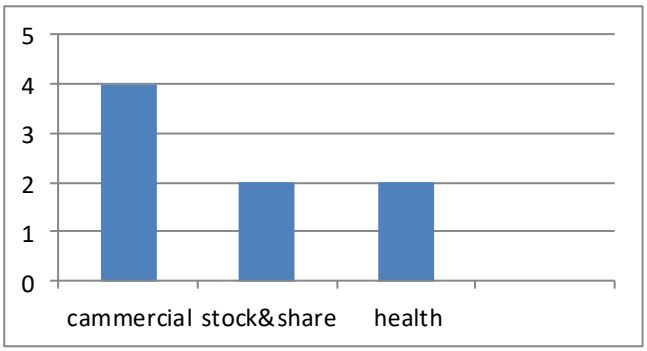


Fig 8: Learning user behavior per Group

VI. CONCLUSION

It is well known that actors in a network demonstrate correlated behaviors. In this work, we aim to predict the outcome of collective behavior given a social network and the behavioral information of some actors. In particular, we explore scalable learning of collective behavior when millions of actors are involved in the network. As existing approaches to extract social dimensions suffer from scalability, it is imperative to address the scalability issue.

To address the scalability issue, we ***propose an edge-centric*** clustering scheme to extract social dimensions and a scalable k-means variant to handle edge clustering. The model based on the sparse social dimensions shows comparable prediction performance as earlier proposed approaches to extract social dimensions. In reality, each edge can be associated with multiple affiliations while our current model assumes only one dominant affiliation.

An incomparable advantage of our model is that it easily scales to handle networks with millions of actors while the earlier models fail. This scalable approach offers a viable solution to effective learning of online collective behavior on a large scale.

Since the proposed *Edge Cluster* 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.

ACKNOWLEDGMENT

There are several people we would like to thank. First, we would like to thank Dr. Kancharla Ramaiah, correspondent and secretary of Prakasam Engineering College,Kandukur, India for his encouragement and support and providing us with the facility for completing this paper.

REFERENCES

1. L. Tang and H. Liu, "Toward predicting collective behavior via social dimension Extraction," IEEE Intelligent Systems, vol. 25, pp. 19-25, 2010.
2. "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.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

3. 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. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.74.036104
4. L. Tang and H. Liu, "Scalable learning of collective behavior based on sparse social dimensions, "in CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge manage ment. New York, NY, USA: ACM, 2009, pp. 1107-1116.
5. P. Single and M. Richardson, "Yes, there is a correlation: - from social networks to personal behavior on the web," in WWW '08: Proceeding of the 17th international conference on World Wide Web. New York,NY, USA: ACM, 2008, pp. 655-664.
6. M. McPherson, L. Smith-Lavin, and J. M. Cook, "Birds of a feather: Homophily in social networks," Annual Review of Sociology, vol. 27, pp. 415-444, 2001. ACM, 2005, pp. 1371-1374.
7. S. A. Macskassy and F. Provost, "Classification in networked data: A toolkit and a univariate case study,"
8. X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in ICML, 2003.