Volume 2, Issue 7, July 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Social Dimension Extraction for Scalable Learning of Collective Behaviour

S.M.Subhani*
Department of CSE
BVSR Engg.College
Chimakurthy, A.P, India

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.

Keywords*— *classification with network data, collective behaviour, 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.

Mostly, we have access to the connectivity information between users, but we have no idea why they are connected to each other. This heterogeneity of connections limits the effectiveness of a commonly used technique — collective inference for network classification. A recent framework based on social dimensions [2] is shown to be effective in ad- dressing this heterogeneity. The framework suggests a novel way of network classification, first, capture the latent affiliations of actors by extracting social dimensions based on network connectivity, and next apply extant data mining techniques to classification based on the extracted dimensions. In the initial study, modularity maximization [3] was employed to extract social dimensions. The superiority of this framework over other representative relational learning methods has been verified with social media data in [2].

The original framework, however, is not scalable to handle networks of colossal sizes because the extracted social dimensions are rather dense. 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.

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].

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. 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 topics simultaneously, or click on multiple types of ads. 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 $Y = \{c1, \cdots, c_K\}$. Given network G = (V, E, Y) where V is the vertex set, E is the edge set and $Yi \subseteq Y$ 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 $V^U = V - V^L$.

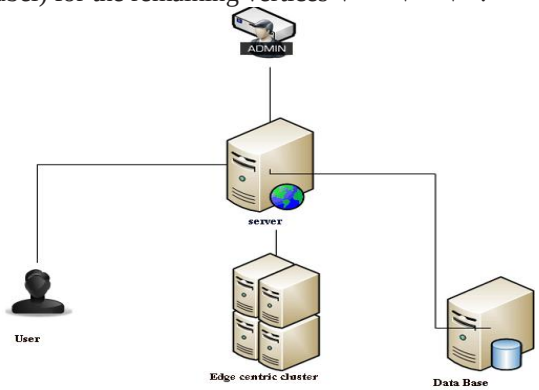

Fig. 1 Architecture for learning collective behaviour

TABLE I: Social Dimension Representation

Actors	Affiliation -1	Affiliation -2	…....Affiliation -k
1	0	1	..…….. 0.8
2	0.5	0.3	……..... 0
.	.		.
.	.		.
.	.		.

III. SPARSE SOCIAL DIMENSIONS

In this section, we first show one toy example to illustrate the intuition of communities in an "edge" view and then present potential solutions to extract sparse social dimensions.

A. Edge-Centric View

Though SocioDim with soft clustering for social dimension extraction demonstrated promising results, its scalability is limited. A network may be sparse (i.e., the density of connectivity is very low), whereas the extracted social dimensions are not sparse. Let's look at the toy network with two communities in Figure 1. Its social dimensions following modularity maximization are shown in Table 2. Clearly, none of the entries is zero. When a network expands into millions of actors, a reasonably large number of social dimensions need to be extracted. The corresponding memory requirement hinders both the extraction of social dimensions and the subsequent discriminative learning. Hence, it is imperative to develop some other approach so that the extracted social dimensions are sparse.

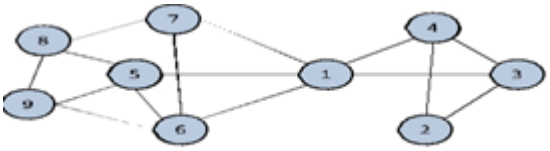

Fig. 2 A Toy Example

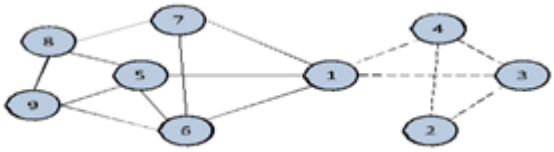

Fig. 3 Edge Clusters

TABLE II: 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

Consider one extreme case that an actor has only one connection. It is expected that he is probably active in only one affiliation. It is not necessary to assign a non-zero score for each of the many other affiliations. Assuming each connection represents one involved affiliation, we can expect the number of affiliations an actor has is no more than that of his connections. For this reason ***we propose an edge-centric view*** rather than defining a community as a set of nodes, we

redefine it as *a set of edges*. Thus, communities can be identified by partitioning edges of a network into disjoint sets. *An actor is considered associated with one affiliation if one of his connections is assigned to that affiliation*.

For instance, the two communities in Figure 2 can be represented by two edge sets in Figure 3, where the dashed edges represent one affiliation, and the remaining edges denote the second affiliation. The disjoint edge clusters in Figure 3 can be converted into the representation of social dimensions as shown in the last two columns in Table 2, where an entry is 1 (0) if an actor is (not) involved in that corresponding social dimension. Node 1 is affiliated with both communities because it has edges in both sets. By contrast, node 3 is assigned to only one community, as all its connections are in the dashed edge set. To extract sparse social dimensions, we partition edges rather than nodes into disjoint sets. The edges of those actors with multiple affiliations (e.g., actor 1 in the toy network) are separated into different clusters.

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:

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

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

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:

$$\frac{\alpha-1}{\alpha-2}\,\frac{1}{k} - \left(\frac{\alpha-1}{\alpha-2} - 1\right) k^{-\alpha+1} \qquad (2)$$

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

- 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 updates the centroids

Fig. 4 Overview of Edge Cluster Algorithm

B. Clustering Edge Instances

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. This results in a typical feature-based data format as in Table 3.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.

TABLE III: Edge Instances of the Toy Network in Figure 1

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
				.....					

By taking into account the two concerns above, we devise a k-means variant as shown in Figure 5.

Input: data instances$\{x_i | 1 \leq i \leq m\}$
Number of clusters k
Output:$\{idx_i\}$

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_{idx}i
14. until change of objective value < ϵ

Fig. 5 Algorithm of Scalable k-means Variant

Similar to k-means, this algorithm also maximizes within cluster similarity as shown in Eq. (3)

$$\arg_s max \sum_{i=1}^{k} \sum_{x_i \in s_i} \frac{x_j \cdot \mu_i}{\|x_j\| \|\mu_i\|} \qquad (3)$$

Where k is the number of clusters, *S* = *{S*1, *S*2. . . *Sk}* is the set of clusters, and *μi* is the centroid of cluster *Si*. In Figure 5,

we keep only a vector of *Marxism* to represent the maximum similarity between one data instance and a centroid. In each iteration, we first identify (edges) are associated with few (much less than *k*) centroids.

By taking advantage of the feature-instance mapping, the cluster assignment for all instances (lines 5-11 in Figure 5) 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 can learn the collective behaviour.
Therefore the collective behavior algorithm shown in fig 6.

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 5.
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. 6 Algorithm for Learning of Collective Behaviour

IV. 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 IV

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
EdgeCluster-100	300.8	3.8M	$1.1\text{x } 10^{-1}$	$2.2\text{x } 10^{-1}$	187	23.5
EdgeCluster-500	357.8	4.9M	$6.0\text{x } 10^{-2}$	$1.1\text{x } 10^{-1}$	344	30.0
EdgeCluster-1000	307.2	5.2M	$3.2\text{x } 10^{-2}$	$6.0\text{x } 10^{-1}$	408	31.8
EdgeCluster-2000	294.6	5.3M	$1.6\text{ x } 10^{-2}$	$3.1\text{x } 10^{-2}$	598	32.4
EdgeCluster-5000	230.3	5.5M	$6\text{ x } 10^{-3}$	$1.3\text{x } 10^{-2}$	682	32.4
EdgeCluster-10000	195.6	5.6M	$3\text{x } 10^{-3}$	$7\text{x } 10^{-2}$	882	33.3

TableV Sparsity Comparison on Flickr Data with 80, 513 Nodes

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

Table VI 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	219
Edge Cluster-1000	779.2	42.3M	2.3×10^{-3}	5.0×10^{-3}	325	232
Edge Cluster-2000	558.9	44.2M	1.2×10^{-3}	2.6×10^{-3}	375	243
Edge Cluster-5000	554.9	45.6M	5.0×10^{-4}	1.0×10^{-3}	253	250
Edge Cluster-10000	561.2	46.4M	2.5×10^{-4}	5.1×10^{-4}	355	254
Edge Cluster-20000	507.5	47.0M	1.3×10^{-4}	2.6×10^{-4}	305	258
Edge Cluster-50000	597.4	48.2M	5.2×10^{-5}	1.1×10^{-4}	297	262

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 4-6. 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 6.

This is due to the efficacy of the proposed k-means variant in Figure 5. 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.

V. CONCLUSION

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. Our approach follows a social dimension based learning framework. Social dimensions are extracted to represent the potential affiliations of actors before discriminative learning occurs. As ***existing approaches*** to extract social dimensions suffer from scalability, it is imperative 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. Essentially, each edge is treated as one data instance, and the connected nodes are the corresponding features. Then, ***the proposed k-means clustering*** algorithm can be applied to partition the edges into disjoint sets, with each set representing one possible affiliation. This model, based on the sparse social dimensions, shows comparable prediction performance with earlier social dimension approaches. 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.

REFERENCES

[1] L. Tang and H. Liu, "Toward predicting collective behavior via social dimensionExtraction," 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.

[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.

