E-ISSN: 2321–9637
Volume 1, Issue 5, December 2013
International Journal of Research in Advent Technology
Available Online at: http://www.ijrat.org

COLLECTIVE BEHAVIOR OF ONLINE USER BASED ON SOCIAL MEDIA

Azeem Ahmad [1], Sujata Deshmukh [2]
M.E. Computer Engineering
Lokmanya Tilak College of Engineering, Navi Mumbai
[1] azeemahmad12345@gmail.com, [2] desh_suja78@yahoo.com

ABSTRACT—
Collective behavior refers to how individuals behave when they are exposed in a social network environment. In this article, we examine how we can predict online behaviors of users in a network, given the behavior information of some actors in the network. Many social media tasks can be connected to the problem of collective behavior prediction. Since connections in a social network represent various kinds of relations, a social-learning framework based on social dimensions is introduced. This framework suggests extracting social dimensions that represent the latent affiliations associated with actors, and then applying supervised learning to determine which dimensions are informative for behavior prediction. It demonstrates many advantages, especially suitable for large-scale networks, paving the way for the study of collective behavior in many real-world applications.

***Keywords*—**Social Networking Sites, Social Dimensions, Relational Learning, Edge-Centric Clustering.

1. INTRODUCTION

This study of collective behavior is to understand how individuals behave in a social networking environment. Oceans of data generated by social media like Facebook, Twitter, Flicker, and YouTube which present opportunities and challenges to study collective behavior on a large scale. In this work, we aim to learn to predict collective behavior in social media. In particular, given information about some individuals, how can we infer the behavior of 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 large size, involving thousands of millions 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. Social media facilitate people of all walks of life to connect to each other.

In the initial study, modularity maximization is exploited to extract social dimensions. With huge number of actors, the dimensions cannot even be held in memory. In this work, we propose an effective edge centric approach to extract sparse social dimensions. The advancement in computing and communication technologies enables people to get together and share information in innovative ways. Social networking sites (a recent phenomenon) empower people of different ages and backgrounds with new forms of collaboration, communication, and collective intelligence.

Sparsifying 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. We prove that with our proposed approach, sparsity of social dimensions is guaranteed.

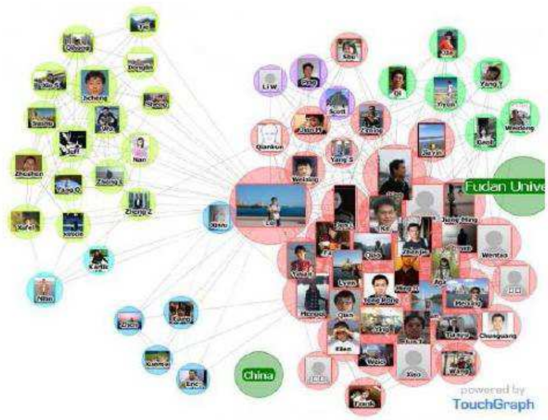

Figure 1: Contacts of One User in Facebook

2. COLLECTIVE BEHAVIOR LEARNING

The recent boom of social media enables the study of collective behavior in a large scale. Here, behavior can include a broad range of actions: join a group, connect to a person, click on some ad, become interested in certain topics, date with people of certain type, etc. When people are exposed in a social network environment, their behaviors are not independent [1, 9]. That is, their behaviors can be influenced by the behaviors of their friends. This naturally leads to behavior correlation between connected users.

This behavior correlation can also be explained by *homophily*. Homophily [4] is a term coined in 1950s to explain our tendency to link up with one another in ways that confirm rather than test our core beliefs. Essentially, we are more likely to connect to others sharing certain similarity with us. This phenomenon has been observed not only in the real world, but also in online systems [3]. Homophily leads to behavior correlation between connected friends. In other words, friends in a social network tend to behave similarly. Take marketing as an example, if our friends buy something, there's better-than-average chance we'll buy it too.

In this work, we attempt to utilize the behavior correlation presented in a social network to predict the collective behavior in social media. Given a network with behavior information of some actors, how can we infer the behavior outcome of the remaining ones within the same network? Here, we assume the studied behavior of one actor can be described with K class labels {c1, · · · , cK}. For each label, ci can be 0 or 1. For instance, one user might join multiple groups of interests, so 1 denotes the user subscribes to one group and 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. That is, the studied behavior can be described by a single label with 1 and 0 denoting corresponding meanings in its specific context, like whether or not one user voted for Barack Obama in the presidential election.

The problem we study can be described formally as follows:

Suppose there are K class labels Y = {c1, … , cK}. Given network A = (V,E, Y) where V is the vertex set, E is the edge set and $Y_i \subseteq \mathcal{Y}$ are the class labels of a vertex $\bar{v}_i \in V$, and given known values of Yi for some subsets of vertices V^L, how to infer the values of Yi (or a probability estimation score over each label) for the remaining vertices

$V^U = V - V^L$?

E-ISSN: 2321–9637
Volume 1, Issue 5, December 2013
International Journal of Research in Advent Technology
Available Online at: http://www.ijrat.org

Note that this problem shares the same spirit as within network classification [5]. It can also be considered as a special case of semi-supervised learning [10] or relational learning [2] when objects are connected within a network. Some of the methods, if applied directly to social media, yield limited success [6], because connections in social media are pretty noise and heterogeneous.

In the next section, we will discuss the connection heterogeneity in social media, briefly review the concept of social dimension, and anatomize the scalability limitations of the earlier model proposed in [6], which motivates us to develop this work.

Table 1: Social Dimension Representation

Actors	Affiliation-1	Affiliation-2	$\cdots$	Affiliation-k
1	0	1	$\cdots$	0.8
2	0.5	0.3	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$

3. SOCIAL DIMENSIONS

Connections in social media are heterogeneous. People can connect to their family, colleagues, college classmates, or some buddies met online. Some of these relations are helpful to determine the targeted behavior (labels) but not necessarily always so true. For instance, Figure 1 shows the contacts of the first author on Facebook. The densely-knit group on the right side is mostly his college classmates, while the upper left corner shows his connections at his graduate school. Meanwhile, at the bottom left are some of his high-school friends. While it seems reasonable to infer that his college classmates and friends in graduate school are very likely to be interested in IT gadgets based on the fact that the user is a fan of IT gadget (as most of them are majoring in computer science), it does not make sense to propagate this preference to his high-school friends. In a nutshell, people are involved in different affiliations and connections are emergent results of those affiliations. These affiliations have to be differentiated for behavior prediction.

However, the affiliation information is not readily available in social media. Direct application of collective inference[5] or label propagation [11] treats the connections in as a social network homogeneously. This is especially problematic when the connections in the network are noisy. To

address the heterogeneity presented in connections, we have proposed a framework (*SocDim*) [6] for collective behavior learning.

The framework *SocDim* 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. One example of the social dimension representation is shown in Table 1. The entries show the degree of one user involving in an affiliation. These social dimensions can be treated as features of actors for the subsequent discriminative learning. Since the network is converted into features, typical classifier such as support vector machine and logistic regression can be employed. The discriminative learning procedure will determine which latent social dimension correlates with the targeted behavior and assign proper weights.

Now let's re-examine the contacts network in Figure 1. One key observation is that when actors are belonging to the same affiliations, they tend to connect to each other as well. It is reasonable to expect people of the same department to interact with each other more frequently. Hence, to infer the latent affiliations, we need to find out a group of people who interact with each other more frequently than random. This boils down to a classical

E-ISSN: 2321–9637
Volume 1, Issue 5, December 2013
International Journal of Research in Advent Technology
Available Online at: http://www.ijrat.org

community detection problem. Since each actor can involve in more than one affiliations, a soft clustering scheme is preferred.
In the instantiation of the framework *SocDim*, modularity maximization [7] is adopted to extract social dimensions. The social dimensions correspond to the top eigenvectors of a modularity matrix. It has been empirically shown that this framework outperforms other representative relational learning methods in social media. However, there are several concerns about the scalability of *SocDim* with modularity maximization:

• The social dimensions extracted according to modularity maximization are dense. Suppose there are 1 million actors in a network and 1, 000 dimensions1 are extracted. Suppose standard double precision numbers are used, holding the full matrix alone requires 1M × 1K × 8 = 8G memory. This large-size dense matrix poses thorny challenges for the extraction of social dimensions as well as the subsequent discriminative learning.

• The modularity maximization requires the computation of the top eigenvectors of a modularity matrix which is of size n × 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.

• Networks in social media tend to evolve, with new members joining, and new connections occurring between existing members each day. This dynamic nature of networks entails efficient update of the model for collective behavior prediction. Efficient online up-date of eigenvectors with expanding matrices remains a challenge.

Consequently, it is imperative to develop scalable methods that can handle large-scale networks efficiently without extensive memory requirement. In the next section, we elucidate an edge-centric clustering scheme to extract *sparse* social dimensions. With the scheme, we can update the social dimensions efficiently when new nodes or new edges arrive in a network.

4. ALGORITHM—EDGECLUSTER

In this section, we first show one toy example to illustrate the intuition of our proposed edge-centric clustering scheme *EdgeCluster*, and then present one feasible solution to handle large-scale networks.

4.1 Edge-centric view

As mentioned earlier, 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, begging for abundant memory space. Let's look at the toy network in Figure 2. The column of modularity maximization in Table 2 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. Hence, it is essential to develop some approach such that the extracted social dimensions are sparse.
The social dimensions according to modularity maximization or other soft clustering scheme tend to assign a non-zero score for each actor with respect to each affiliation. However, it seems reasonable that the number of affiliations one user can participate in is upperbounded by the number of
connections. 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 nonzero score for each affiliation. Assuming each connection represents one dominant affiliation, we expect the number of affiliations of one actor is no more than his connections.

E-ISSN: 2321–9637
Volume 1, Issue 5, December 2013
International Journal of Research in Advent Technology
Available Online at: http://www.ijrat.org

Instead of directly clustering the nodes of a network into some communities, we can take 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 3. Based on the features (connected nodes) of each edge, we can cluster the edges into two sets as in Figure 4, 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 4 can be converted into the social dimensions as the last two columns for edge-centric clustering in Table 2. Actor 1 is involved in both affiliations under this *EdgeCluster* scheme.

In summary, to extract social dimensions, we cluster edges rather than nodes in a network into disjoint sets. To achieve this, k-means clustering algorithm can be applied. The edges of those actors involving in multiple affiliations (e.g., actor 1 in the toy network) are likely to be separated into different clusters. Even though the partition of edge-centric view is disjoint, the affiliations in the node-centric view can overlap. Each actor can be involved in multiple affiliations.

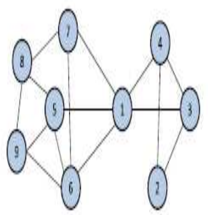

Figure 2: A Toy Example

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

Figure 3: Edge-Centric View

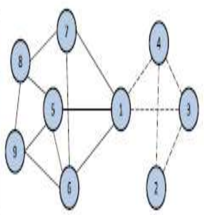

Figure 4: Edge Clusters

Actors	Modularity Maximization	Edge-Centric Clustering	
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

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

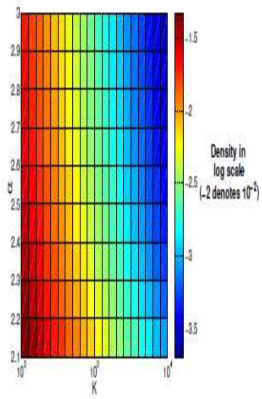

Figure 5: Density Upperbound of Social Dimensions

In addition, the social dimensions based on edge-centric clustering are *guaranteed to be sparse*. This is because the affiliations of one actor are no more than the connections he has. Suppose we have a network with m edges, n nodes and k social dimensions are extracted. Then each node vi has no more than min(di, k) non-zero entries in its social dimensions, where di is the degree of node vi. We have the following theorem.

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 extracted based on edge-centric clustering is bounded by the following formula:*

$$
\begin{aligned}
density &\leq \frac{\sum_{i=1}^{n} \min(d_i, k)}{nk} \\
&= \frac{\sum_{\{i|d_i \leq k\}} d_i + \sum_{\{i|d_i > k\}} k}{nk} \qquad (1)
\end{aligned}
$$

Moreover, for networks in social media where the node degree follows a power law distribution, the upper bound in Eq. (1) can be approximated as follows:

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

Note that the upperbound in Eq. (1) is network specific whereas Eq.(2) gives an approximate upperbound for a family of networks. It is observed that most power law distributions occurring in nature have $2 \leq \alpha \leq 3$ [8]. Hence, the bound in Eq. (2) is valid most of the time. Figure 5 shows the function in terms of α and k. Note that when k is huge (close to 10,000), the social dimensions becomes extremely sparse $(< 10^{-3})$. In reality, the extracted social dimensions are typically even sparser than this upperbound as shown in later experiments. Therefore, with edge-centric clustering, the extracted social dimensions are sparse, alleviating the memory demand and facilitating efficient discriminative learning in the second stage.

4.2 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. That is, m = O(n). This can be verified via the properties of power law distribution. Suppose a network with n nodes follows a power law distribution as

$$p(x) = Cx^{-\alpha}, \quad x \geq x_{min} > 0$$

where α is the exponent and C is a normalization constant.

```
Input: data instances {x_i | 1 ≤ i ≤ m}
       number of clusters k
Output: {idx_i}
construct a mapping from features to instances
initialize the centroid of cluster {C_j | 1 ≤ j ≤ k}
repeat
  Reset {MaxSim_i}, {idx_i}
  for j=1:k
    identify relevant instances S_j to centroid C_j
    for i in S_j
      compute sim(i, C_j) of instance i and C_j
      if sim(i, C_j) > MaxSim_i
        MaxSim_i = sim(i, C_j)
        idx_i = j;
  for i=1:m
    update centroid C_{idx_i}
until no change in idx or change of objective < ε
```

Figure 6: Algorithm for Scalable K-means Variant

Then the expected number of degree for each node is [8]:

$$E[x] = \frac{\alpha - 1}{\alpha - 2}\, x_{min}$$

where x_{min} is the minimum nodal degree in a network. In reality, we normally deal with nodes with at least one connection, so $x_{min} >= 1$. The α of a real-world network following power law is normally between 2 and 3 as mentioned in [8]. Consider a network in which all the nodes have non-zero degrees, the expected number of edges is

$$E[m] = \frac{\alpha - 1}{\alpha - 2}\, x_{min} \cdot n/2$$

Unless α is very close to 2, in which case the expectation diverges, the expected number of edges in a network is linear to the total number of nodes in the network.

Still, millions of edges are the norm in a large-scale social network. Direct application of some existing k-means implementation cannot handle the problem. E.g., the k-means code provided in Matlab package requires the computation of the similarity matrix between all pairs of data instances, which would exhaust the memory of normal PCs in seconds. Therefore, implementation with an online fashion is preferred.

On the other hand, the edge data is quite sparse and structured. As each edge connects two nodes in the network, the corresponding data instance has exactly only two non-zero features as shown in Figure 3. This sparsity can help accelerate the clustering process if exploited wisely. We conjecture that the centroids of k-means should also be feature-sparse. Often, only a small portion of the data instances share features with the centroid. Thus, we only need to compute the similarity of the centroids with their relevant instances. In order to efficiently identify the instances relevant to one centroid, we build a mapping from the features (nodes) to instances (edges) beforehand. Once we have the mapping, we can easily identify the relevant instances by checking the non-zero features of the centroid. By taking care of the two concerns above, we thus have a k-means variant as in Figure 6

E-ISSN: 2321–9637
Volume 1, Issue 5, December 2013
International Journal of Research in Advent Technology
Available Online at: http://www.ijrat.org

to handle clustering of many edges. We only keep a vector of MaxSim to represent the maximum similarity between one data instance with a centroid. In each iteration, we first identify the set of relevant

Input: network data, labels of some nodes
Output: labels of unlabeled nodes

1. convert network into edge-centric view as in Figure 3
2. perform clustering on edges via algorithm in Figure 6
3. construct social dimensions based on edge clustering
4. build classifier based on labeled nodes' social dimensions
5. use the classifier to predict the labels of unlabeled ones based on their social dimensions

Figure 7: Scalable Learning of Collective Behavior

instances to a centroid, and then compute the similarities of these instances with the centroid. This avoids the iteration over each instance and each centroid, which would cost O(mk) otherwise. Note that the centroid contains one feature (node) if and only if any edge of that node is assigned to the cluster. In effect, most data instances (edge) 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 6) can be fulfilled in O(m) time. To compute the new centroid (lines 12-13), it costs O(m) time as well. Hence, each iteration costs O(m) time only. Moreover, the algorithm only requires 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 the edges into disjoint sets. Later as we show, even for a network with millions of actors, this clustering can be finished in tens of minutes while modularity maximization becomes impractical.

As a simple k-means is adopted to extract social dimensions, it is easy to update the social dimensions if the network changes. If a new member joins a network and a new connection emerges, we can simply assign the new edge to the corresponding clusters. The update of centroids with new arrival of connections is also straightforward. This k-means scheme is especially applicable for dynamic large scale networks.

In summary, to learn a model for collective behavior, we take the edge-centric view of the network data and partition the edges into disjoint sets. Based on the edge clustering, social dimensions can be constructed. Then, discriminative learning and prediction can be accomplished by considering these social dimensions as features. The detailed algorithm is summarized in Figure 7.

ACKNOWLEDGEMENT

I pay my gratitude to Lokmanya Tilak College of Engineering & IEEE and my guide Prof. Sujata Deshmukh.

References

[1] M. Hechter. *Principles of Group Solidarity*. University of California Press, 1988.
[2] L. Getoor and B. Taskar, editors. *Introduction to Statistical Relational Learning*. The MIT Press, 2007.
[3] A. T. Fiore and J. S. Donath. Homophily in online dating: when do you like someone like yourself? In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1371–1374, 2005.
[4] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001.

E-ISSN: 2321–9637
Volume 1, Issue 5, December 2013
International Journal of Research in Advent Technology
Available Online at: http://www.ijrat.org

[5] S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *J. Mach. Learn. Res.*, 8:935–983, 2007.
[6] L. Tang and H. Liu. Relational learning via latent social dimensions. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 817–826, 2009.
[7] M. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 74(3), 2006.
[8] M. Newman. Power laws, Pareto distributions and Zipf's law. *Contemporary physics*, 46(5):323–352, 2005.
[9] G. L. Zacharias, J. MacMillan, and S. B. V. Hemel, editors. *Behavioral Modeling and Simulation: From Individuals to Societies*. The National Academies Press, 2008.
[10] X. Zhu. Semi-supervised learning literature survey. 2006.
[11] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields andharmonic functions. In *ICML*, 2003.