# COLLECTIVE BEHAVIOUR IN SOCIAL NETWORKING

**A.Saritha[1]**

[1]*Student,CSE, Prakasm Engineering College, A.P., India,* ***saritha@gmail.com***

**Abstract**

*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, Flickr, and YouTube 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 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 nonscalable methods.*

***Index Terms:*** *Classification with network data, collective behavior, community detection, social dimensions.*

---------- *** ----------

## 1. INTRODUCTION

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. Prodigious numbers of online volunteers collaboratively write encyclopedia articles of unprecedented scope and scale; online marketplaces recommend products by investigating user shopping behavior and interactions; and political movements also exploit new forms of engagement and collective action. In the same process, social media provides ample opportunities to study human interactions and collective behavior on an unprecedented scale. 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 weinfer 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.

Typically, the connections in social media networks are not homogeneous. Different connections are associated with distinctive relations. For example, one user might maintain connections simultaneously to his friends, family, college classmates, and colleagues. This relationship information, however, is not always fully available in reality. 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 addressing 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. 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 [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.

## 2. COLLECTIVE BEHAVIOR

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 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 homophily [6]. Homophily is a term coined in the 1950s toexplain our tendency to link with one another in ways that confirm, rather than test, our core beliefs. Essentially, we are more likely to connect to others who share certain similarities with us. This phenomenon has been observed not only in the many processes of a physical world, but also in online systems [7], [8]. Homophily results in behavior correlations between connected friends. In other words, friends in a social network tend to behave similarly.

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 $\{c_1, \ldots, c_k\}$. Each label, $c_i$, can be 0 or 1. For instance, one user might join multiple groups of interest, so $c_i = 1$ denotes that the user subscribes to group i, and $c_i = 0$ 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 1 and 0. For example, if the event is the presidential election, 1 or 0 indicates whether or not a voter voted for Barack Obama.

It should be noted that this problem shares the same spirit as within-network classification [9]. It can also be considered as a special case of semi-supervised learning [10] or relational learning [11] where objects are connected within a network. Some of these methods, if applied directly to social media, yield only limited success [2]. This is because connections in social media are rather noisy and heterogeneous. In the next section, we will discuss the connection heterogeneity in social media, review the concept of social dimension, and anatomize the scalability limitations of the earlier model proposed in [2], which provides a compelling motivation for this work.

## 3 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 [9] or label propagation [12] 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.

**TABLE 1**
**Social Dimension Representation**

| Actors | Affiliation-1 | Affiliation-2 | ⋯ | Affiliation-*k* |
|---|---|---|---|---|
| 1 | 0 | 1 | ⋯ | 0.8 |
| 2 | 0.5 | 0.3 | ⋯ | 0 |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |

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

A key observation is that actors of the same affiliation tend to connect with each other. For instance, it is reasonable to expect people of the same department to interact with each other more frequently. Hence, to infer actors' latent affiliations, we need to find out a group of people who interact with each other more frequently than at random. This boils down to a classic community detection problem. Since each actor can get involved in more than one affiliation, a soft clustering scheme is preferred.

In the initial instantiation of the framework SocioDim, a spectral variant of modularity maximization [3] 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 on social media data. However, there are several concerns about the scalability of SocioDim with modularity maximization:

- Social dimensions extracted according to soft clustering, such as modularity maximization and probabilistic methods, are dense. Suppose there are 1 million actors in a network and 1,000 dimensions are extracted. If standard double precision numbers are used, holding the full matrix alone requires 1 M X 1 K X 8 = 8 G memory. This large-size dense matrix poses thorny challenges for the extraction of social dimensions as well as subsequent discriminative learning.
- Modularity maximization requires us to compute the top eigenvectors of a modularity matrix, which is the same size as a given network. In order to extract k communities, typically k - 1 eigenvectors are computed. For a sparse or structured matrix, the eigenvector computation costs $O(h(mk + nk^2 + k^3))$ time [13], where h, m, and n are the number of iterations, the number of edges in the network, and the number of nodes, respectively. Though computing the top single eigenvector (i.e., k = 1), such as
- PageRank scores, can be done very efficiently, computing thousands of eigenvectors or even more for a mega-scale network 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 an efficient update of the model for collective behavior prediction. Efficient online updates of eigenvectors with expanding matrices remain a challenge.

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.

## 4. 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.

### 4.1. Communities in an 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 Fig. 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.

It seems reasonable to state that the number of affiliations one user can participate in is upper bounded by his 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 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. 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.

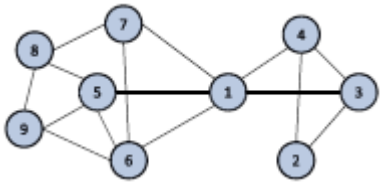


Fig. 1. A Toy example.

TABLE 2
Social Dimension(s) of the Toy Example

| Actors | 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 |

An actor is considered associated with one affiliation if one of his connections is assigned to that affiliation. For instance, the two communities in Fig. 1 can be represented by two edge sets in Fig. 2, where the dashed edges represent one affiliation, and the remaining edges denote the second affiliation. The disjoint edge clusters in Fig. 2 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. Though the partition in the edge view is disjoint, the affiliations in the node-centric view can overlap. Each node can engage in multiple affiliations.

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 $\min(d_i,k)$ nonzero entries in her social dimensions, where di is the degree of node $v_i$.

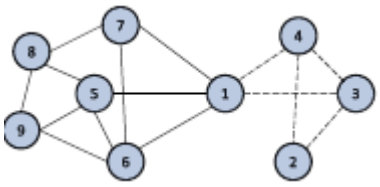


Fig. 2. Edge clusters.

### 4.2 Edge Partition via Line Graph Partition

In order to partition edges into disjoint sets, one way is to look at the "dual" view of a network, i.e., the line graph [15]. We will show that this is not a practical solution. In a line graph L(G), each node corresponds to an edge in the original network G, and edges in the line graph represent the adjacency between two edges in the original graph. The line graph of the toy example is shown in Fig. 3. For instance, e(1, 3) and e(2, 3) are connected in the line graph as they share one terminal node 3.

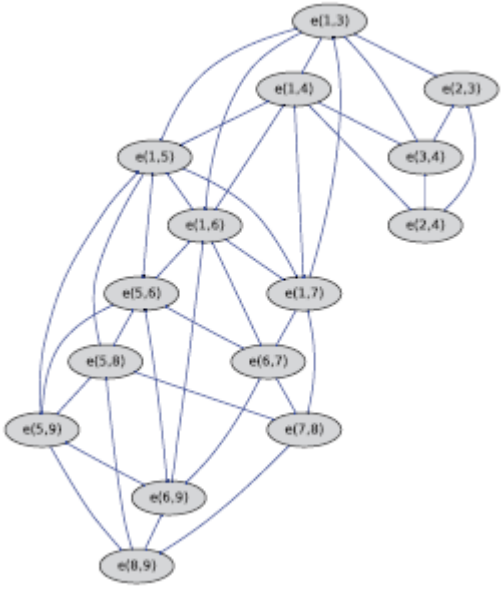


Fig. 3. The line graph of the toy example in Fig. 1. Each node in the line graph corresponds to an edge in the original graph.

Given a network, graph partition algorithms can be applied to its corresponding line graph. The set of communities in the line graph corresponds to a disjoint edge partition in the original graph. Recently, such a scheme has been used to detect overlapping communities. It is, however, prohibitive to construct a line graph for a megascale network. We notice that edges connecting to the same node in the original network form a clique in the corresponding line graph. For example, edges e(1,3), e(2, 3), and e(3, 4) are all neighboring edges of node 3 in Fig. 1. Hence, they are adjacent to each other in the line graph in Fig. 3, forming a clique. This property leads to many more edges in a line graph than in the original network.

### 4.3 Edge Partition via Clustering Edge Instances

In order to partition edges into disjoint sets, we treat edges as data instances with their terminal nodes as features. For instance, we can treat each edge in the toy network in Fig. 1 as one instance, and the nodes that define edges as features. This results in a typical feature-based data format as in Table 3. Then, a typical clustering algorithm like k-means clustering can be applied to find disjoint partitions.

TABLE 3
Edge Instances of the Toy Network in Fig. 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 |
| ⋮ | | | | | ..... | | | | |

One concern with this scheme is that the total number of edges might be too huge. Owing 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).

Still, millions of edges are the norm in a large-scale network. Direct application of some existing k-means implementation cannot handle the problem. For example, the k-means code provided in the 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, an implementation with online computation is preferred.

On the other hand, the data of edge instances are quite sparse and structured. As each edge connects two nodes, the corresponding edge instance has exactly only two nonzero features as shown in Table 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. Hence, we just need to compute the similarity of the centroids with their relevant instances. In order to efficiently identify instances relevant to one centroid, we build a mapping from features (nodes) to instances (edges) beforehand. Once we have the mapping, we can easily identify the relevant instances by checking the nonzero features of the centroid.

**Input:** data instances $\{x_i|1 \le i \le m\}$
number of clusters $k$
**Output:** $\{idx_i\}$

1. construct a mapping from features to instances
2. initialize the centroid of cluster $\{C_j|1 \le j \le k\}$
3. **repeat**
4. Reset $\{MaxSim_i\}$, $\{idx_i\}$
5. **for** j=1:k
6. identify relevant instances $S_j$ to centroid $C_j$
7. **for** $i$ in $S_j$
8. compute $sim(i, C_j)$ of instance $i$ and $C_j$
9. **if** $sim(i, C_j) > MaxSim_i$
10. $MaxSim_i = sim(i, C_j)$
11. $idx_i = j$;
12. **for** i=1:m
13. update centroid $C_{idx_i}$
14. **until** change of objective value $< \epsilon$

Fig. 4. Algorithm of scalable k-means variant.

By taking into account the two concerns above, we devise a k-means variant as shown in Fig. 4. Similar to kmeans, this algorithm also maximizes within cluster similarity. And k is the number of clusters, S = {$S_1$, $S_2$, . . . , $S_k$} is the set of clusters, and $\mu_i$ is the centroid of cluster Si. In Fig. 4, we keep only a vector of MaxSim to represent the maximum similarity between one data instance and a centroid. In each iteration, we first identify the instances relevant to a centroid, and then compute similarities of these instances with the centroid. This avoids the iteration over each instance and each centroid, which will 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 (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 Fig. 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 featureinstance 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.

### 4.4 Regularization on Communities

The extracted social dimensions are treated as features of nodes. Conventional supervised learning can be conducted. In order to handle large-scale data with high dimensionality and vast numbers of instances, we adopt a linear SVM, which can be finished in linear time [18]. Generally, the larger a community is, the weaker the connections within the community are. Hence, we would like to build an SVM relying more on communities of smaller sizes by modifying the typical SVM objective function.

**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. 5. Algorithm for learning of collective behavior.

In summary, we conduct the following steps to learn a model for collective behavior. Given a network (say, Fig. 1), we take the edge-centric view of the network data (Table 3) and partition the edges into disjoint sets (Fig. 2). Based on the edge clustering, social dimensions can be constructed (Table 2). Certain regularizations can be applied. Then, discriminative learning and prediction can be accomplished by considering these social dimensions as features. The detailed algorithm is summarized in Fig. 5.

## 5. CONCLUSIONS AND FUTURE WORK

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. 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. With this edgecentric view, we show that the extracted social dimensions are guaranteed to be sparse. 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.

In social media, multiple modes of actors can be involved in the same network, resulting in a multimode network. For instance, in YouTube, users, videos, tags, and comments are intertwined with each other in coexistence. 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 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.

## ACKNOWLEDGEMENT

We would like to express our sincere thanks to Sri. Dr. Kancharla Ramaiah Secretary and Correspondent, Prakasam Engineering College, Kandukur, A.P. India for his support with providing research environment . We are extremely thankful to our collogues , friends and family members who are cooperated in this work.

## REFERENCES

[1] L. Tang and H. Liu, “Toward Predicting Collective Behavior via Social Dimension Extraction,” IEEE Intelligent Systems, vol. 25, no. 4, pp. 19-25, July/Aug. 2010.

[2] L. Tang and H. Liu, “Relational Learning via Latent Social Dimensions,” KDD ’09: Proc. 15th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining, pp. 817-826, 2009.

[3] M. Newman, “Finding Community Structure in Networks Using the Eigenvectors of Matrices,” Physical Rev. E (Statistical, Nonlinear, and Soft Matter Physics), vol. 74, no. 3, p. 036104, http:// dx.doi.org/10.1103/PhysRevE.74.036104, 2006.

[4] L. Tang and H. Liu, “Scalable Learning of Collective Behavior Based on Sparse Social Dimensions,” CIKM ’09:

Proc. 18th ACM Conf. Information and Knowledge Management, pp. 1107-1116, 2009.

[5] P. Singla and M. Richardson, "Yes, There Is a Correlation: - From Social Networks to Personal Behavior on the Web," WWW '08: Proc. 17th Int'l Conf. World Wide Web, pp. 655-664, 2008.

[6] M. McPherson, L. Smith-Lovin, and J.M. Cook, "Birds of a Feather: Homophily in Social Networks," Ann. Rev. of Sociology, vol. 27, pp. 415-444, 2001.

[7] A.T. Fiore and J.S. Donath, "Homophily in Online Dating: When Do You Like Someone Like Yourself?," CHI '05: Proc. CHI '05 Extended bstracts on Human Factors in Computing Systems, pp. 1371-1374, 2005.

[8] H.W. Lauw, J.C. Shafer, R. Agrawal, and A. Ntoulas, "Homophily in the Digital World: A LiveJournal Case Study," IEEE Internet Computing, vol. 14, no. 2, pp. 15-23, Mar./Apr. 2010.

[9] S.A. Macskassy and F. Provost, "Classification in Networked Data: A Toolkit and a Univariate Case Study," J. Machine Learning Research, vol. 8, pp. 935-983, 2007.

[10] X. Zhu, "Semi-Supervised Learning Literature Survey," technical report, http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey_12_9_2006.pdf, 2006.

[11] Introduction to Statistical Relational Learning, L. Getoor and B. Taskar, eds. The MIT Press, 2007.

[12] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," Proc. Int'l Conf. Machine Learning (ICML), 2003.

[13] S. White and P. Smyth, "A Spectral Clustering Approach to Finding Communities in Graphs," Proc. SIAM Data Mining Conf. (SDM), 2005.

[14] M. Newman, "Power Laws, Pareto Distributions and Zipf's Law," Contemporary Physics, vol. 46, no. 5, pp. 323-352, 2005.

[15] F. Harary and R. Norman, "Some Properties of Line Digraphs," Rendiconti del Circolo Matematico di Palermo, vol. 9, no. 2, pp. 161- 168, 1960.

[16] T. Evans and R. Lambiotte, "Line Graphs, Link Partitions, and Overlapping Communities," Physical Rev. E, vol. 80, no. 1, p. 16105, 2009.

[17] Y.-Y. Ahn, J.P. Bagrow, and S. Lehmann, "Link Communities Reveal Multi-Scale Complexity in Networks," http://www. citebase.org/abstract?id=oai:arXiv.org:0903.3178, 2009.

[18] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification," J. Machine Learning Research, vol. 9, pp. 1871-1874, 2008.

[19] J. Hopcroft and R. Tarjan, "Algorithm 447: Efficient Algorithms for Graph Manipulation," Comm. ACM, vol. 16, no. 6, pp. 372-378, 1973.

[20] J. Neville and D. Jensen, "Leveraging Relational Autocorrelation with Latent Group Models," MRDM '05: Proc. Fourth Int'l Workshop Multi-Relational Mining, pp. 49-55, 2005.